TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

VYUŽITÍ HLUBOKÝCH NEURONOVÝCH SÍTÍ V SYSTÉMECH ROZPOZNÁVÁNÍ ŘEČI

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Martin Paroubek**
Vedoucí práce: Ing. Petr Červa, Ph.D.

Liberec 2014

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

THE USE OF DEEP NEURAL NETWORKS IN SPEECH RECOGNITION SYSTEMS

Diploma thesis

Study programme: N2612 – Electrical Engineering and Informatics
Study branch: 1802T007 – Information Technology

Author: **Bc. Martin Paroubek**
Supervisor: Ing. Petr Červa, Ph.D.

Liberec 2014

Tento list nahraďte
originálem zadání.

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Poděkování

Rád bych na tomto místě poděkoval panu Ing. Petru Červovi, Ph.D. za cenné rady, připomínky, ochotu a čas věnovaný konzultacím mé diplomové práce a také panu Ing. Ladislavu Šepsovi za uvedení do problematiky a četné konzultace. V neposlední řadě bych chtěl poděkovat rodině a přátelům, kteří mi byli oporou.

Abstrakt

Práce se zabývala využitím nového hybridního systému DNN-HMM pro rozpoznávání řeči. V teoretické části byla představena základní problematika rozpoznávání řeči a neuronových sítí. Na základě těchto informací bylo možné představit hluboké neuronové sítě a jejich propojení s HMM systémem. Z důvodu velkého množství různých parametrů pro trénování neuronových sítí proběhla rešerše existujících postupů a jejich výsledků, kterými byla inspirována praktická část.

Cílem praktické části bylo prozkoumat vliv uspořádání neuronové sítě, vliv předtrénování a vliv velikosti trénovacího korpusu na přesnost rozpoznávání. Na základě těchto výsledků byl vytvořen akustický model, který byl porovnán se současným systémem pro rozpoznávání řeči GMM-HMM.

Trénování neuronových sítí probíhalo na GPU použitím modifikovaných skriptů knihovny Theano. Následné vyhodnocení bylo provedeno pomocí vlastních skriptů. K dispozici byl trénovací korpus s 56 hodinami polské řeči a vytvořené modely byly odzkoušeny na 3 testovacích sadách obsahujících publicistický a odborný styl. K porovnání výsledků byla použita tzv. accuracy.

Celkem bylo takto vytvořeno více než 250 akustických modelů, které se také lišily dobou trénování, neboť kriterium ukončení trénování je stále předmětem zkoumání. Celkem doba k jejich natrénování zabrala více než 62 dní. Bylo zjištěno, že využití neuronových sítí, jakožto akustických modelů, přináší několikaprocentní zlepšení oproti současnému systému a zároveň také že diskriminativní předtrénování nemá žádný vliv na přesnost sítě. Dále byla popsána topologie s nejvyšší přesností a bylo zjištěno, že vliv množství dat v trénovacím korpusu může být závislý na kontextu testovací sady.

Klíčová slova: akustický model, rozpoznávání řeči, hluboké neuronové sítě, LVCSR, DNN-HMM

Abstract

The thesis dealt with using a new DNN-HMM system for speech recognition. In its theoretical part a basic outline of speech recognition and neural networks was laid out. Based on this information, it was possible to introduce deep neural networks and their connections with the HMM system. Due to the immense number of various parameters used when training neural networks, a research was conducted concerning the procedures already in use and their results. The practical part was inspired by this research.

The goal of the practical part was to establish the level of influence of the neural network's layout, its pretraining and the size of its training data set on the accuracy of its recognition. Based on the results, an acoustic model was created, which was then compared to the GMM-HMM speech recognition system currently in use.

The training of this neural network was conducted on a GPU using modified scripts from the Theano library. The subsequent evaluation used the author's original scripts. There was a training data set available with 56 hours of Polish speech and the resulting models were tested on three sets, which contained texts in journalistic and professional style. The so-called "accuracy" was used to compare the results.

More than 250 acoustic models were created in this way. The models differed from one another in the amount of time required for training, as this is still under research. In total, all the models required more than 62 days to finish training. It was established that using neural networks as acoustic models results in an improvement of several percent over the system currently in use and futhermore that pretraining using the discriminative method has no effect on the network's accuracy. Topology was described with the highest degree of accuracy and it was concluded that the amount of data present in the training data set may be dependent on the context of the testing set.

Keywords: acoustic model, speech recognition, deep neural networks, LVCSR, DNN-HMM

Obsah

Seznam obrázků

Seznam symbolů a zkratek

AE-BN	Autoencoder Bottleneck
BMMI	Boosted Maximum Mutual Information
BMVS	Bing Mobile Voice Search Application
CUDA	Compute Unified Device Architecture
DBN	Deep Belief Network
DNN	hluboké neuronové sítě
DT	diskriminativně trénované
EM	Expectation Maximization
GMM	gaussovské mixturové modely
GPU	grafický procesor
HLDA	Heteroscedastic Linear Discriminant Analysis
HMM	Skryté Markovovy modely
ITE	ústav informačních technologií a elektroniky
LDA	Linear Discriminant Analysis
LVCSR	Large Vocabulary Continuous Speech Recognition
MFCC	Mel-Frequency Cepstral Coefficients
PLP	Perceptual Linear Predictive Coefficients
RBM	Restricted Boltzmann Machine
ReLU	Rectified Linear Units
SAT	Speaker Adaptive Training
SLM	Stochastické jazykové modely
TIMIT	korpus foneticky a lexikálně přepsané řeči v aj
VT	Vocal Tract
VTLN	Vocal Tract Length Normalization
WER	Word Error Rate

1 Úvod

Neuronové sítě v posledních letech zažívají velký boom. Dokonce v reakci na první vítězství hlubokého učení resp. neuronových sítí v soutěži Merck, byl v roce 2012 publikován článek [13] magazínem New York Times. Neuronové sítě byly představeny tak, že se snaží napodobit to, jak mozek pracuje s novými informacemi a jak se z nich učí. Dále v článku byl citován zajímavý výrok Dr. Rashida, který v té době dohlížel na organizaci Microsoft's worldwide research. K hlubokému učení uvedl, že je to největší dramatická změna v úspěšnosti rozpoznávání řeči od roku 1979 a zároveň zmínil, že tento pokrok byl možný mimo jiné díky vzestupu grafických procesorů. V jiné části článku byl také uveden úspěch laboratoře Swiss A. I. z univerzity v Luganu, jejíž program pro rozpoznávání německých dopravních značek překonal i lidského experta.

Hluboké neuronové sítě jsou ve světě jednou z aktuálně zkoumaných metod, přestože byly poprvé zmíněny před více jak 70lety. V posledních pěti letech bylo zjištěno, že přinášejí zlepšení úspěšnosti v rozpoznávání v mnoha odvětvích včetně akustického modelování. V článku [3] bylo napsáno, že neexistuje důvod věřit, že jsou v současnosti používány optimální typy skrytých neuronů, nebo optimální typy topologie neuronových sítí. Také je velmi pravděpodobné, že předtrénovací a trénovací algoritmy můžou být modifikovány tak, aby se redukovalo přetrénování a množství výpočtů. Je očekáváno, že rozdíl v úspěšnostech systémů, které používají DNN a systému s GMM, poroste ve prospěch DNN.

V laboratoři počítačového zpracování řeči na Technické univerzitě se tento nový přístup zatím nevyzkoušel. V České republice v době psaní této práce nebyly vydány žádné články nebo práce, které by se zabývaly takovou tématikou. Tudíž v této práci byly zpracovány jedny z prvotních experimentů a je nutné uvést, že celková problematika použití hlubokých neuronových sítí pro rozpoznávání řeči a jejich napojení na současné metody přesahuje rámec diplomové práce.

Trénovat modely neuronových sítí se považuje za náročné. Např. v článku [11] to bylo dokonce označeno za hlavní překážku, neboť k rozumnému nastavení množství různých parametrů, které jsou mnohdy na sobě závislé, je potřeba mít nejen dostatek

zkušeností, ale i „zručnosť“. Jejich náhodné zkoušení je výpočetně náročné např. v této práci trvalo trénování jednotlivých modelů neuronových sítí od 12 hodin do 7 dnů.

V první kapitole byly nejdříve představeny základní stavební prvky neuronových sítí model neuronu a perceptron. V práci byl použit pro rozpoznávání řeči hybridní systém složený z neuronových sítí a současného přístupu. Z tohoto důvodu bylo nutné představit i základní principy rozpoznávání řeči, aby bylo možné v následující kapitole popsat, která část byla hlubokými neuronovými sítěmi nahrazena. Ve čtvrté kapitole byla provedena rešerše zveřejněných experimentů, jimiž bylo inspirováno zadání práce. Následující kapitola je věnována praktické části práce. Nejdříve byl představen toolkit pro trénování, následně bylo uvedeno nastavení experimentů a provedena analýza dosažených výsledků.

Cíle práce

- Seznamte se s problematikou hlubokých neuronových sítí a s metodami jejich učení.
- Na pracovní stanici s výkonnou GPU zprovozněte softwarový balík Theano umožňující trénování hlubokých neuronových sítí.
- Pomocí vlastních skriptů pro přípravu a zpracování trénovacích a testovacích dat a pomocí daného softwarového nástroje natrénujte různé typy akustických modelů využívající neuronové sítě. Experimentujte s různou topologií sítě, použitým postupem pro trénování a s množstvím dat.
- Experimentálně ověřte přesnost vytvořených akustických modelů, proveďte analýzu dosažených výsledků a porovnejte je s výsledky získanými za pomoci klasických technik, popřípadě s výsledky popsanými v literatuře.

Míra hodnocení

K porovnání úspěšnosti rozpoznávání byla použita míra accuracy [6] dále nazývána přesnosti:

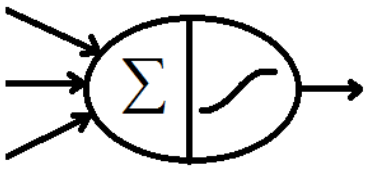
$$accuracy = \frac{N - D - S - I}{N} \cdot 100, \quad (1)$$

kde N je celkový počet slov, D je počet vynechaných slov, S je počet chyb rozpoznaných slov a I je počet vložených slov.

2 Základy neuronových sítí

V kapitole byly přestaveny základy neuronových sítí, které byly nutné k pochopení významu a analýze experimentů v praktické části.

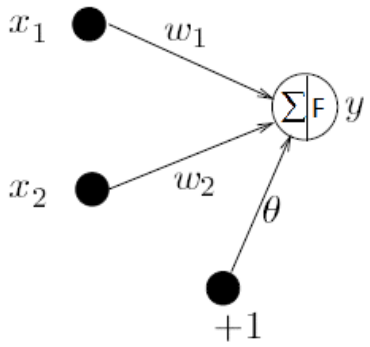
Základní stavební jednotkou každé sítě je neuron, viz obrázek 2.1. Do neuronu vstupují orientované hrany nazývané synaptická spojení. Každá z těchto hran je ohodnocena takzvanou váhou, která je jediná proměnlivá část neuronové sítě a působí jako paměť. Váhy se trénují opakovaným předkládáním trénovacích dat, které se skládají z příznaků a jejich příslušnosti ke třídám. Příznaky jsou číselné parametry objektů. Úkolem neuronu je každý vstup vynásobit váhou a následně všechny tyto mezivýsledky sečíst. Ve druhé fázi se na součet aplikuje aktivační funkce, která byla popsána později v kapitole 2.3.4. Neurony lze spojovat sériově, potom se jedná o vrstvy neuronové sítě a v případě paralelního spojení se definuje šířka vrstvy neuronové sítě.



Obrázek 2.1: Model neuronu

2.1 Perceptronová síť

Russell a Norvig [2] představili nejzákladnější model neuronové sítě tzv. perceptronovou síť, kterou lze označit i jako jednovrstevnou dopřednou síť. Skládá se z jedné vrstvy výstupních neuronů. V této vrstvě může být jeden nebo více neuronů. Synaptická spojení jsou orientována od vstupů směrem k výstupům. Speciálním případem perceptronové sítě je perceptron, který má právě jeden výstup viz obrázek č. 2.2.



Obrázek 2.2: Perceptron

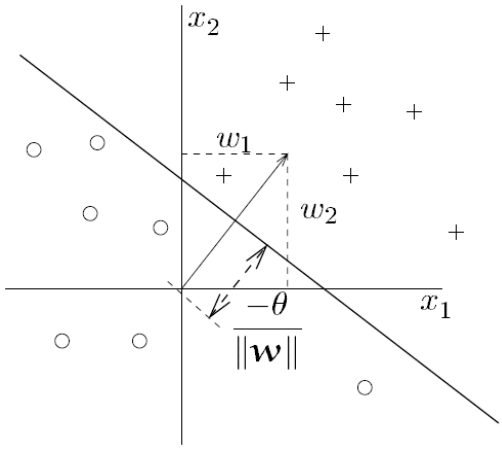
(převzato z [1])

Kröse a Smagt [1] popsali výstup perceptronu na obrázku 2.2 rovnicí 2, ve které parametr w_i představuje váhu x_i vstupní parametry, θ práh (angl. bias term) a F aktivační funkci.

$$y = F\left(\sum_{i=1}^{2} w_i x_i + \theta\right) \tag{2}$$

Váhy w_i definují sklon přímky, oddělující při klasifikaci dvě třídy, a θ definuje její posun resp. jak daleko je přímka od bodu počátku viz obrázek 2.3.

V knize [2] bylo uvedeno, že perceptronová síť může reprezentovat pouze lineárně separovatelné problémy. To například znamená, že ve dvourozměrném prostoru existuje přímka taková, která oddělí třídy koleček od třídy plusů viz obrázek 2.3. Mezi lineárně separovatelné problémy patří například logické funkce AND, OR a NOT, ale nepatří mezi ně XOR, ten tudíž nelze řešit klasickou perceptronovou sítí.

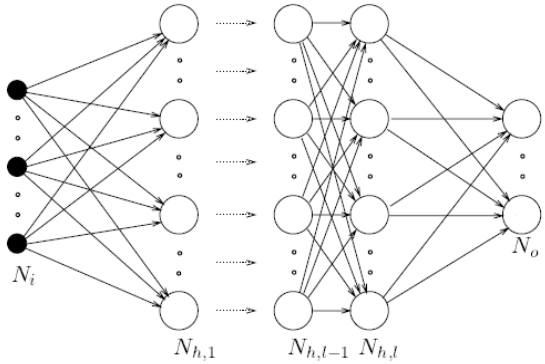


Obrázek 2.3: Geometrická reprezentace diskriminativní funkce a vah

(Převzato z [1])

2.2 Vícevrstevný perceptron

Vícevrstevným perceptronem se podle knihy [1] nazývá vícevrstevná dopředná sít, která už umí reprezentovat i nelineární problémy resp. s každou další vrstvou dokáže řešit komplexnější problémy. Skládá se viz obrázek 2.4 z N_i vstupních neuronů, z *l* skrytých vrstev o šířce *h* neuronů a N_o výstupních neuronů. Pro výstupy neuronů platí stejná rovnice 2 jako pro perceptron, který byl popsán v předchozí kapitole. Ve vícevrstvém perceptronu neexistují žádná přímá spojení mezi vstupními a výstupními vrstvami ani žádná spojení mezi neurony ve stejné vrstvě. Počet výstupních neuronů může být různý od počtu vstupních neuronů a může se také lišit od počtu neuronů ve skrytých vrstvách.



Obrázek 2.4: Vícevrstevný perceptron s l vrstvami.

(Převzato z [1])

Russell a Norvig [2] napsali, že je jednoduché spočítat chybu pro výstupní neurony, ale pro neurony ve skrytých vrstvách je to o mnoho obtížnější. Nejspíše protože pro trénovací sadu jsou výstupy známé, ale to se nedá říct o výstupech skrytých neuronů. Zároveň také zmínili neefektivitu učících algoritmy pro vícevrstevné sítě, které negarantují konvergenci ke globálnímu minimu resp. k modelu sítě, který poskytne nejlepší možný výsledek. Také uvedli nevýhody neuronových sítí spočívající v poměrně dlouhé době učení a v obtížném řešení problému lokálních minim. Nejpoužívanější metodou pro učení vícevrstevných sítí je algoritmus zpětného šíření chyby, který byl představen v roce 1969 autory Bryson a Ho a byl popsán v kapitole 2.3.1.

2.3 Trénování neuronové sítě

V této podkapitole byl představen základní postup pro trénování neuronových sítí. Předlohou pro postup trénování, pro algoritmus zpětného šíření chyby a jednotlivé rovnice byla kniha [7].

K trénování neuronových sítí se používá metoda největšího spádu (angl. gradient descent) viz rovnice 3, pomocí které je vypočítána hodnota váhy w v následující iteraci n+1. Ta se vypočítává z hodnoty váhy w v iteraci n a přičtením úpravy vah $\Delta w_{ji}(n)$, která je vypočítána pomocí delta pravidla viz rovnice 8 anebo pomocí zobecněného delta pravidla viz rovnice 13.

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) \tag{3}$$

Pro výpočet $\Delta w_{ji}(n)$ je nejprve nutné spočítat chybový signál $e_j(n)$, který je dán rozdílem očekávaného výstupu $d_j(n)$ od skutečného výstupu neuronové sítě $y_j(n)$ viz rovnice 4**.**

$$e_j(n) = d_j(n) - y_j(n) \tag{4}$$

Z chybového signálu $e_j(n)$ se vypočítá chybová funkce $\xi(n)$ definována rovnicí 5 za předpokladu, že je neuronová síť trénována po vzorcích. Výpočet se provádí přes všechny neurony C ve výstupní vrstvě. Tato chyba se poté pomocí algoritmu zpětného šíření distribuuje mezi jednotlivé neurony resp. váhy.

$$\xi(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \tag{5}$$

2.3.1 Algoritmus zpětného šíření chyby

Russel a Norvig [2] napsali o algoritmu zpětného šíření chyby (angl. Backpropagation), že byl ignorován do půlky osmdesátých let a že jedním z možných důvodů mohla být náročnost na výpočetní kapacitu u netriviálních problémů. Kröse [1] zmínil hlavní ideu algoritmu, a totiž že chyby pro neurony

ve skrytých vrstvách můžou být rozhodnuty zpětným šířením chyb neuronů z výstupní vrstvy.

Obrázek 2.5 znázorňuje neuron j, který se nachází v iteraci n. Do neuronu j vstupují výstupy neuronů $y_i(n)$ z předešlé vrstvy i, v iteraci n a práh y_0. Vážený součet $v_j(n)$ vstupů $y_i(n)$ s váhami $w_{ji}(n)$ byl definován rovnicí 6. Aktivační funkce φ_j poté upraví $v_j(n)$, jejíž aplikací se spočítá výstup neuronu $y_j(n)$ viz rovnice 7.

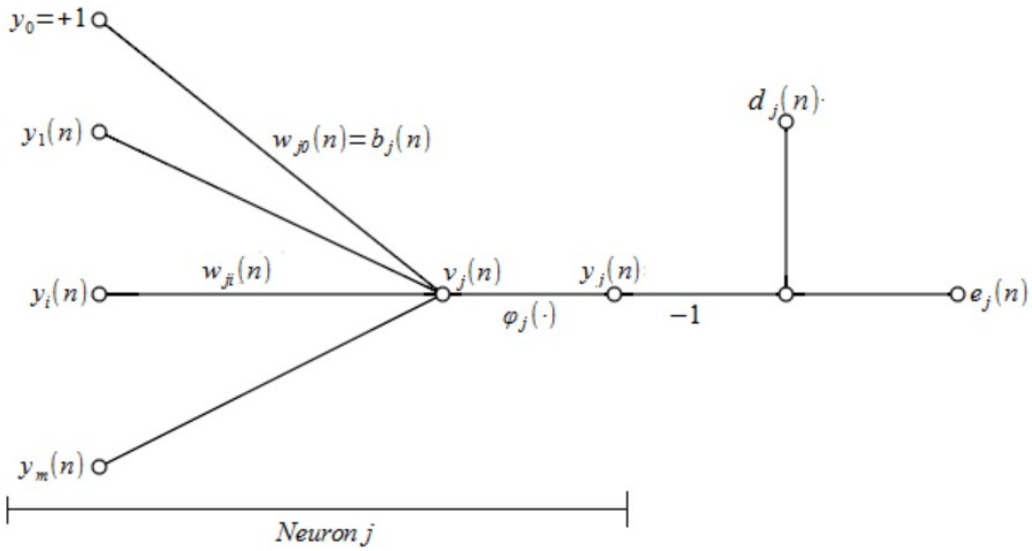
$$v_j(n)=\sum_{i=0}^{m} w_{ji}\, y_i(n) \qquad \textbf{(6)}$$

$$y_j(n)=\varphi_j(v_j(n)) \qquad \textbf{(7)}$$

Úprava vah $\Delta w_{ji}(n)$ je potom definována jako součin parametru učení η, lokálního gradientu $\delta_j(n)$ a výstupu neuronu $y_i(n)$ viz rovnice 8. Využitím delta pravidla viz rovnice 3 se spočítá váha pro další iteraci.

$$\Delta w_{ji}(n)=\eta\delta_j(n)\, y_i(n) \qquad (8)$$

Výpočet lokálního gradientu $\delta_j(n)$ se definuje různě pro neurony ve výstupní a skryté vrstvě.



Obrázek 2.5: Model neuronu a výpočtu chybového signálu

Převzato z [7]

Případ 1: Neuron j je ve výstupní vrstvě

Pro výstupní vrstvu jsou známy očekávané výsledky, proto se použije vzorec 4 pro výpočet chybového signálu $e_j(n)$, který se dosadí do vzorce 9 pro výpočet lokálního gradientu. Nejdříve je nutné spočítat derivaci aktivační funkce $\varphi_j'(v_j(n))$ viz rovnice 10, kde $y_j(n)$ výstup neuronu j.

$$\delta_j(n) = \frac{\partial \xi(n)}{\partial v_j(n)} = e_j(n)\varphi_j'(v_j(n)) \tag{9}$$

$$\varphi'(v_j(n)) = \frac{\partial y_j(n)}{\partial v_j(n)} \tag{10}$$

Případ 2: Neuron j je ve skryté vrstvě

Pro skryté vrstvy nejsou známy očekávané výsledky, z tohoto důvodu se lokální gradient počítá rekurzivně podle vzorce 11, ve kterém δ_j představuje lokální gradient neuronu j ve skryté vrstvě, δ_k lokální gradient neuronu ve vrstvě následující (směrem k výstupní), w_{kj} váhu z neuronu j do neuronu k.

$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \tag{11}$$

2.3.2 Podrobněji k chybové funkci

Při výpočtu chybové funkce se v knize [7] rozlišuje učení po jednotlivých vzorcích, které bylo použito při vysvětlení algoritmu zpětného šíření chyby, a nebo učení po dávkách, které bylo použito při trénování**.** V takovém případě jsou váhy neuronové sítě změněny po ukončení jedné epochy trénování, resp. po využití všech trénovacích dat. Chybovou funkci $\xi_{av}(n)$ je nutné spočítat přes všechny již aplikované vzorky N v jednotlivých iteracích (viz rovnice 12). Parametr $e_j(n)$ značí chybový signál v iteraci n.

$$\xi_{av}(n) = \frac{1}{2} N \sum_{n=1}^{N} \sum_{j \in C} e_j^2(n) \tag{12}$$

Metoda učení po dávkách se dobře paralelizuje, jejímž výsledkem je přesnější odhad vektoru gradientu, čímž je garantována konvergence metody největšího spádu k lokálnímu minimu. Pro každou epochu je nutné trénovací vzorky promíchat.

Oproti tomu metoda učení po jednotlivých vzorcích se špatně paralelizuje, ale umožňuje se vyhnout uváznutí v lokálním minimu, neboť se pohybuje v prostoru vah s různými směry. Dávkové učení využívá lépe redundantnost trénovacích dat a je schopné v nich vysledovat malé změny, zejména když je prostředí zodpovědné za generování tzv. nestacionární dat (šum, rušení).

2.3.3 Rychlost učení

Haykin [7] uvedl, že algoritmus zpětného šíření chyby provádí přiblížení k trajektorii (v prostoru vah), která byla vypočítána pomocí metody největšího spádu. Čím menší se zvolí učící parametr η, tím menší jsou změny vah a tím je přesnější (hladší) trajektorie. Nevýhodou je, že se tím také zpomalí proces učení. Pokud je parametr η příliš velký (z důvodů zrychlení učení), tak může dojít k nestabilitě vah (např. oscilace). Jednoduchou metodou zrychlení procesu učení a vyhnutí se nebezpečí nestability je použití zobecněného delta pravidla, které navíc zahrnuje tzv. momentum viz rovnice 13.

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n), \tag{13}$$

kde momentum α je většinou kladné číslo, a $\Delta w_{ji}(n-1)$ značí úpravu váhy z minulé iterace. Zbytek rovnice byl vysvětlen v předchozí kapitole. Tímto rozšířením je možné se vyhnout mělkým lokálním minimům.

2.3.4 Aktivační funkce

Aktivační funkce, v některé literatuře též nazývány přenosové nebo přechodové funkce, upravují výstup z neuronu. Je důležité, aby byly diferencovatelné, neboť jinak není možné použít algoritmus zpětného šíření chyby. Také musí být nelineární, jinak by použití vícevrstevných sítí ztratilo smysl, protože by neměly větší modelovací možnosti než sítě perceptronové. V práci byla použita aktivační funkce sigmoida (viz rovnice 13). Ta představuje rostoucí nelineární funkci, jejímž výstupem jsou

hodnoty v rozmezí od nuly do jedné. Zároveň omezuje rozsah hodnot, tudíž nedojde k přetečení datového typu.

$$y = \frac{1}{1 + e^{-x}} \tag{14}$$

Ve výstupních vrstvách neuronových sítí se používá funkce softmax. Slouží ke klasifikaci do více tříd a vypočítává podmíněnou pravděpodobnost P (třída| data). Postup výpočtu byl uveden v rovnici 14, ve které x_j značí vstup neuronu j a k je index přes všechny třídy, resp. přes všechny neurony ve výstupní vrstvě.

$$p_j = \frac{e^{x_j}}{\sum_k e^{x_k}} \tag{15}$$

Mezi další alternativy aktivačních funkcí, které se používají ve skrytých vrstvách neuronových sítí, patří hyperbolický tangens s výstupními hodnotami v rozmezí mínus jedna až jedna. Novinkou jsou neurony nazývané ReLU (rectified linear units), které používají aktivační funkci max (0,x). Na základě experimentů ve článku [14] bylo zjištěno, že se s ní rychleji trénuje nejen pro nižší výpočetní náročnost, ale i kvůli rychlejší konvergenci k lokálnímu minimu, a že také umí lépe zobecňovat příznaky.

2.4 Krosvalidace

Podle knihy [7] spočívá krosvalidace, neboli holdout metoda, v náhodném rozdělení dat, které jsou určeny k učení, na trénovací vzorek a testovací sadu. Trénovací vzorek je potom rozdělen na dvě části:

- odhadní část určená k vytvoření modelu,

- validační část určená k testování nebo validaci modelu.

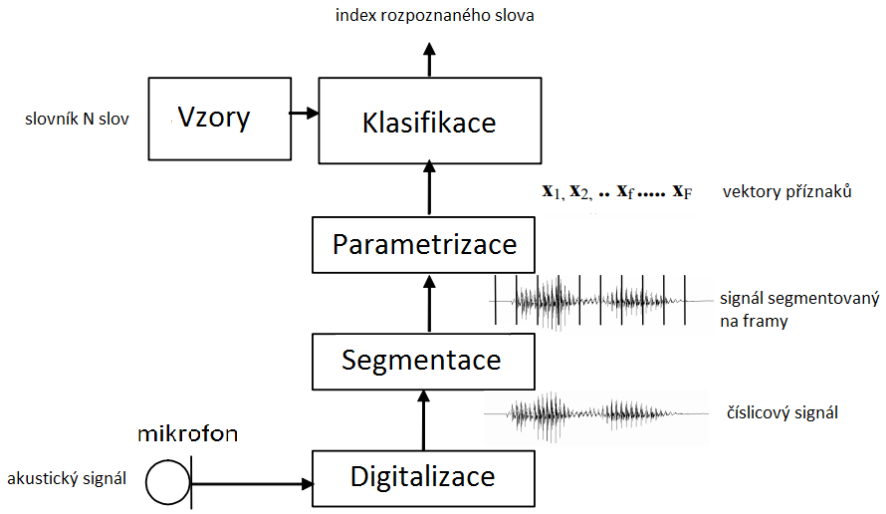
Důvodem je zkontrolovat model na jiných datech, různých od těch, které byly použity k trénování parametrů. Přesto může dojít k přetrénování na validační části, aby se tomu předešlo, používá se k vyhodnocení úspěšnosti modelu testovací vzorek dat.

Jiný typ krosvalidace (angl. multifold cross-validation) se používá při nedostatku dat. V takovém případě se data rozdělí na N přibližně stejných částí. Při trénování se vynechá jedna část a ta bude určena k validaci. Pomocí zbytku dat se síť natrénuje. Tento postup se opakuje N krát, přičemž je pokaždé vynechána jiná část. Hodnocení modelu se vypočítává jako průměr druhých mocnin chyb pro všechna opakování.

3 Základní principy rozpoznávání řečí

V práci se pracovalo s hybridním systémem, který vycházel ze současně používaného systému některými zdroji označovaného zkratkou GMM-HMM. Proto bude v kapitole představen současný princip rozpoznávání řeči.

Nouza [5] popsal základní princip rozpoznávání řečí za pomocí obrázku 3.1. Na vstupu je analogový signál snímaný mikrofonem a na výstupu se objevuje informace o přiřazení slova k jedné ze slovníkových položek. Jednotlivé fáze zpracování tvoří digitalizace signálu, jeho segmentace a následná parametrizace a klasifikace. Převod z analogového signálu do číslicové podoby zajišťuje zvuková karta. Číslicový signál se rozděluje na rámce (z angl. frame) o délce 10 ms až 25 ms. Délka se volí tak, aby byla porovnatelná s trváním nejkratší hlásky. Jednotlivé rámce se potom popisují pomocí příznaků, které musí splňovat několik požadavků. Měly by umožnit dostatečné odlišení rámců reprezentujících různé akustické a řečové jevy, zároveň by měly být zvoleny tak, aby pokud možno potlačovaly rysy signálu, které jsou svázány s individualitou řečníka (např. výška a síla hlasu).



Obrázek 3.1: Struktura rozpoznávacího systému pro izolovaně pronášená slova.

Převzato z [6]

3.1 Skryté Markovovy modely

V knize [5] bylo řečeno, že nejvíce dnešních systémů k rozpoznávání řeči používá skryté Markovovy modely, dále jen HMM, pro vypořádání se s dočasnou variací řeči a gaussovské mixturové modely, dále jen GMM, k určení, jak dobře každý stav každého HMM odpovídá příznakovým vektorům reprezentující jednotlivé rámce resp. akustický vstup.

V případě rozsáhlejších slovníků či u aplikací s proměnnými soubory slov se přístup rozpoznávání s celoslovními modely stává neefektivní nebo nepoužitelný. Řešením je přechod od celoslovních modelů k modelům nižších jednotek (např. hlásky neboli fonémy), jejichž výhodou je malý počet (od 20 do 50 v závislosti na jazyce). Oproti tištěným znakům jsou fonémy mnohem variabilnější, neboť jejich zvuková realizace závisí na okolních fonémech. V praxi to znamená, že vytvořit dobré modely fonému je obtížnější než v případě slov.

Modely fonému se trénují na záznamech řeči obsahujících tisíce realizací každé hlásky v různých slovech. Trénovací databáze musí být rozsáhlá (desítky hodin záznamů) a musí obsahovat přesný fonetický přepis všech problémů. K modelování fonému se nejčastěji používá třístavový HMM. Aby bylo možné dobře postihnout akustickou variabilitu, mívají výstupní funkce tvar vícesložkových gaussovských rozloženích. Pokud je hláska reprezentována jediným modelem, bývá tento model označován jako monofon. Pro lepší modelování variability hlásek se používají trifony, které závisí na okolí fonému, a tudíž je vytvořeno pro jedno hlásku více modelů.

3.2 Jazykový a akustický model

Jazykový model podstatně vylepšuje výkon automatického rozpoznávače spojité řeči. Drábková a Nejedlová [5] napsaly, že úkolem jazykového modelu je stanovit omezení a nalézt pravidla, pomocí nichž lze ze slov vytvořit větu. Omezení a pravidla vycházejí z vlastností konkrétního jazyka a mohou být modelována stochastickými či nestochastickými metodami. Stochastické jazykové modely, dále jen SLM, používají pro jazykové modelování pravděpodobností přístup. Jazykové modely přiřazují každé posloupnosti slov W = {w1, w2, w3, …, wn} pravděpodobnost p(W), která je vypočítána z trénovacích dat. Nejrozšířenějším SLM je n-gramový jazykový model. Podle řetězového pravidla pravděpodobnosti platí:

$$
\begin{aligned}
p(W) &= p(w_1, w_{2,}.., w_n) \\
&= p(w_1)\, p(w_2|w_1)\, p(w_3|w_1, w_2) \ldots p(w_n|w_1, w_2 \ldots w_n) \\
&= \prod_{i=1}^{n} p(w_i|w_1, w_2, \ldots., w_{i-1})
\end{aligned}
\tag{16}
$$

V praxi se nejčastěji používá bigramový nebo trigramový jazykový model. Jestliže se pravděpodobnosti vyjádří pomocí četností, které se získají z trénovacích dat, pak pro bigram platí:

$$
p(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n)}{C(w_{n-1})}, \tag{17}
$$

$C(w_{n-1})$ je počet výskytů slova w_{n-1} a $C(w_{n-1},w_n)$ je počet výskytů dvojce slov w_{n-1} a w_n. Hodnoty pravděpodobností jsou pro velké množství dat velice malé, proto se počítá s logaritmy pravděpodobností.

Základní (nevyhlazený) bigramový jazykový model je matice, jejíž hodnoty jsou podmíněnou pravděpodobností určenou pro všechny možné dvojice sousedních slov obsažené v trénovacích datech. Nulová hodnota se v matici může objevit v případě, že se daná posloupnost slov v trénovacím korpusu neobjevila. Protože se ale může objevit v testovací sadě, používají se metody vyhlazování jazykového modelu.

Drábková a Nejedlová [5] také popsaly metodu vyhlazování Witten-Bell Discounting založenou na myšlence, že neviděná $p(w_n|w_{n-1})$ jsou přímo úměrná počtu druhů slov, které byly pozorovány pouze jednou v trénovacím korpusu za slovem w_{n-1}. Nevýhodou vyhlazování tímto způsobem je, že nesprávně rozpočítává pravděpodobnosti u takových dvojic slov w_{n-1} a w_n, kde počet různých dvojic slov s předchůdcem w_{n-1} je vyšší než polovina všech slov ve slovníku. Pro takové případy bývá vhodnější použít jinou vyhlazovací metodu.

4 Hluboké neuronové sítě

V této kapitole byl představen nový trend v akustickém modelování a také jak se propojil se současným systémem.

Hluboké neuronové sítě, dále jen DNN, představují vícevrstevný perceptron s více než jednou vrstvou skrytých neuronů. Každý skrytý neuron obvykle používá jako aktivační funkci tzv. sigmoidu (viz kapitola 2.3.4).

Haykin [7] uvedl, že skryté neurony se chovají jako detektory příznaků. Jak postupuje proces učení, objevují se charakterizující příznaky, které se vytvářejí pomocí nelineárních transformací vstupních dat do příznakového prostoru. V tomto prostoru se může stát, že třídy jsou jednodušeji separovatelné než v originálním prostoru vstupních dat.

DNN mohou být trénované diskriminativně použitím algoritmu zpětného šíření chyby viz kapitola 2.3.1. V článku [3] bylo uvedeno, že neurony na výstupu jsou převedeny do posteriorních pravděpodobností aplikováním aktivační funkce softmax (místo sigmoidy). Pro větší trénovací sady je většinou efektivnější spočítat chybovou funkci na menší náhodné dávce („minibatch“) vybrané z trénovací sady, než to počítat pro celou trénovací sadu. Prahy lze považovat za váhy přicházející z neuronů s výstupem vždy jedna. Pro DNN jsou sousední vrstvy plně propojeny a váhy se inicializují náhodně malými čísly tak, aby se předešlo případu, že by přechody mezi různými dvojicemi neuronů byly stejné.

Hinton a spol. [3] uvedli, že DNN s velkým množstvím skrytých vrstev a neuronů na vrstvu jsou velmi ohebné modely s velkým množstvím parametrů. To je dělá schopné vymodelovat složité a nelineární vztahy mezi vstupy a výstupy. Tato schopnost je důležitá pro vysokou kvalitu akustického modelovaní. Ale pokud se nezvolí počáteční rozsahy vah opatrně, tak zpětně šířené přechody můžou mít různé magnitudy v různých vrstvách. V některých případech mohou špatně zvolené vzorky vést k přetrénování, které může být redukováno pomocí penále vah nebo zastavením trénování předčasně, ale za cenu ztráty další modelovací síly. Další možností k redukci přetrénování při zachování modelovací síly je použití velkých trénovacích dat, ale takové trénování je výpočetně drahé.

Velmi velké GMM akustické modely je podle článku [3] možné trénovat použitím paralelismu na clusterech, ale je těžší využít potenciálu clusterů pro trénování neuronových sítí. V současnosti nejefektivnější metoda je paralelizace maticových operací použitím GPU. Tato metoda poskytne zrychlení o jeden až dva řády (10x až 100x). DNN oproti GMM lépe využívají data, tudíž nepotřebují tolik dat pro to, aby dosáhly obdobných výsledků jako GMM. DNN také získávají více informací z okolních rámců signálu a z modelování svázaných kontextově závislých stavů.

DNN s více vrstvami lze těžko optimalizovat. Proces trénování z náhodného počátečního stavu nemusí být nejlepší cestou, jak najít dobrou sadu vah. Druhou cestou je použít generativní předtrénování vah a poté trénovat DNN diskriminativně.

4.1 Generativní předtrénování

Namísto navrhování detektorů příznaků k tomu, aby dobře rozlišovaly mezi třídami, lze je nejprve navrhnout, aby byly dobré v modelování struktury ve vstupních datech. Základní myšlenka byla uvedena v článku [3]. Síť je učena po jedné vrstvě, jejíž výstupy jsou použity pro trénování následující vrstvy. Po tomto generativním předtrénování mohou být vrstvy spojeny a použity jako lepší startovací pozice pro diskriminativní fázi trénování, během které jsou váhy DNN upravovány menšími kroky. Některé z příznaků vytvořené generativním předtrénováním ve vrstvách vzdálenějších od vstupu nemusí být použitelné pro diskriminaci, ale další mohou být lepší než obyčejné vstupy. Generativní předtrénování najde region v prostoru vah, který umožní diskriminativnímu trénování udělat rychlejší pokrok, a také významně redukuje přetrénování.

Dahl a spol. [4] označili DBN (deep beliefs networks) dopřednými neuronovými sítěmi, které byly generativně předtrénovány za pomocí RBM (Restricted Boltzman Machines). RBM si lze představit jako bipartitní graf mezi viditelnými a skrytými uzly. Viditelné uzly představují pozorování a jsou spojeny se skrytými uzly, které se učí reprezentovat příznaky za pomocí neorientovaných ohodnocených hran. Nejjednodušším případem RBM je binární typ, ve kterém jsou skryté i viditelné uzly binární a stochastické. K vypořádání se s daty, které jsou ohodnoceny z oboru reálných čísel, se podle článku [8] používají tzv. Gaussian-Bernoulli RBM. Váhy RBM se trénují metodou contrastive divergence popsanou v práci [5].

V článku [3] byl shrnut význam předtrénování, které je užitečné zejména v hlubokých neuronových sítí než v mělkých. Obzvláště pokud je k dispozici limitované množství zarovnaných dat. Předtrénování redukuje přetrénování a redukuje čas potřebný pro diskriminativní trénování, což bylo hlavní překážkou v devadesátých letech pro použití neuronových sítí namísto GMM.

4.2 Alternativní metody předtrénování

Předtrénování DNN jako generativních modelů vede k lepším rozpoznávacím výsledkům na úloze TIMIT a také na různých variacích LVCSR úloh. Bylo zjištěno [4], že DNN je možné postupně vymodelovat vrstvu po vrstvě. Nejprve se diskriminativně natrénuje neuronová síť o jedné skryté vrstvě. Potom se vloží nová vrstva mezí skrytou vrstvu a výstupní vrstvu a síť se znova diskriminativně natrénuje. Takto se pokračuje až do požadovaného počtu skrytých vrstev. Poté následuje diskriminativní trénování celé sítě. Tento typ diskriminativního předtrénování funguje dobře v praxi a blíží se výsledkům generativního předtrénování. Další zlepšení je možné dosáhnout zastavením diskriminativního předtrénování předčasně, např. po jedné epoše.

4.3 Další využití DNN pro rozpoznávání řeči

Dosud byla popisována metoda, ve které je GMM nahrazeno DNN akustickým modelem, a tím se vytvoří hybridní DNN-HMM systém, v němž jsou výstupem DNN posteriorní pravděpodobnosti HMM, čímž je nahrazen výstup GMM modelu. Další z možností využití DNN je metoda nazývaná tandem approach [4]. Tato metoda spočívá v extrakci příznakových vektorů z neuronové sítě s úzkým hrdlem, tedy s užší prostřední skrytou vrstvou, jejíž výstupy neuronů slouží jako příznaky pro GMM k modelování.

Méně přímou metodou [4] k vyprodukování příznakových vektorů pro GMM je nejdříve natrénování neuronové sítě o 6 skrytých vrstvách, každá s 1024 neurony a s například 384 výstupnímu neurony. Tato síť nemá hrdlo, ale její vstup do neuronů ve výstupní vrstvě se komprimuje, např. na 40 hodnot pomocí autoenkodéru 384-128-40-384. Tato metoda se nazývá AE-BN.

4.4 Propojení DNN s HMM

Problematika propojeni DNN s HMM byla popsána v článku [4] na straně 6. Po diskriminativním trénování je výstupem DNN posteriorní pravděpodobnost kontextově závislého stavu z HMM za podmínky, že známe příznaky rámce z akustického vstupu p(stav|příznaky). Pro výpočet pomocí Viterbiho algoritmu uvnitř HMM frameworku je potřeba znát věrohodnost p(příznaky|stav). Posteriorní pravděpodobnosti mohou být převedeny na věrohodnost použitím Bayesovy věty.

$$p(příznaky \mid stav) = \frac{p(stav \mid příznaky) \cdot p(příznaky)}{p(stav)} \tag{18}$$

V článku [4] byl postup výpočtu zjednodušen a proběhl vydělením posteriorní pravděpodobnosti četností kontextově závislého stavu v trénovacím korpusu. Dále bylo uvedeno, že pravděpodobnost stavu nemá žádný efekt, a tudíž může být pominuta.

4.5 Současné poznatky

Nejnovější poznatky shrnuty v článku [11] zvyšují náskok DNN-HMM systémů oproti GMM-HMM a neustále se objevují další nové metody a přístupy. Některé z objevů byly zmíněny v této kapitole:

- Bylo zjištěno, že nekorelované příznaky MFCC zhoršují přesnost rozpoznáváním a proto se zkouší různé korelované příznaky například banky filtrů.
- Metody DNN závislé na řečníkovi poskytují malé zlepšení oproti univerzálním podobám.
- Jednou z výhod DNN je možnost rozpoznání i zašuměných signálů nebo poškozených pásek.
- Diskriminativní trénování v případě většího množství zarovnaných akustických dat se projevuje lépe než se myslelo, a tudíž generativní předtrénování může mít nižší vliv.

- Použití plného propojení mezi vrstvami je jednoduché, ale nevýhodné. DNN použité pro akustické modelování pracují lépe, pokud se použije jedna nebo více vrstev, které sdílí váhy skrz blízké frekvence, a sdruží tím pádem odpovědi do podobných frekvencí. Důvodem je vokální trakt, který je u každé osoby různý.
- DNN-HMM systémy mají vyšší úspěšnost při nastavení výstupu DNN na kontextově závislé HMM stavy (použití více rámců pro trénování).
- Použití neuronů s aktivační funkcí sigmoidy bylo překonáno, pokud se použijí ReLU. Tyto neurony sice podléhají rychleji k přetrénování, ale nová regulační metoda dropout je efektivní v jejich kontrolování.
- Regulační metoda dropout náhodně vynechá během trénování některé části neuronové sítě v každé skryté vrstvě. Díky tomu jsou odstraněny skryté neurony, které jsou užitečné pouze v kontextu jiných neuronů.
- DNN architektury lze použít k multi-task učení v několika různých směrech a jsou efektivnější než GMM v použití dat z jedné úlohy ke zlepšení výkonnosti úlohy jiné. Například možnost trénování DNN na více jazycích najednou nebo natrénovat DNN na jednom jazyku a poté na jiném. Microsoft a Google také experimentují s vícejazyčným rozpoznáváním řeči.
- Trénování velmi hlubokých neuronových sítí pomocí metody největšího spádu může být obtížnější, neboť hodnota gradientu se většinou snižuje, jakmile je předána skrz několik vrstev DNN. Řešení se našlo v adaptování parametru učení pro každou vrstvu nebo váhu.

5 Rešerše experimentů

V této kapitole bylo představeno několik experimentů, které sloužily nejen jako motivace, ale i jako podklad pro praktickou část práce.

5.1 TIMIT

TIMIT je trénovací korpus pro vyzkoušení nových přístupů v rozpoznávání řeči. Trénovací sada je malá, aby umožnila vyzkoušet množství variací nových metod a pro většinu technik již existující zveřejněné výsledky. Proto bývá doporučeno nejdříve nový přístup odzkoušet na tomto korpusu a porovnat jeho výsledky s ostatními přístupy. Na druhou stranu se vyskytly případy, kdy zlepšení v úspěšnosti na úloze TIMIT nebylo zlepšením úspěšnosti na úloze LVSR, které může mít nižší požadavky na kvalitu nahrávek a její modely se tvoří z většího množství dat. Přesto TIMIT poskytuje dobrý startovní bod pro vytvoření nového přístupu, speciálně takového, který vyžaduje mnoho výpočtů což DNN bezpochyby jsou.

V článku [8] bylo napsáno, že v případě nastavení jakékoliv kombinace parametrů neuronové sítě – počet skrytých vrstev od 2 do 8, počtu neuronů 1024, 2048 nebo 3072 a počet rámců akustických dat 11, 15 nebo 17 – může mít přesnost do 2 procent od nejlepší kombinace. Z těchto informaci se také vycházelo při nastavení parametrů v praktické části práce.

5.1.1 Předzpracování akustických dat pro DNN

K problematice předzpracování akustických dat se vyjádřil Hinton a spol. [3]. Za state-of-the-art je v současné považován GMM-HMM systémem. Ten ale nevyužívá příznaky vytvořené pomocí bank filtrů, protože jsou silně korelované, tudíž jejich vymodelovaní je náročnější než v případě MFCC příznaků, které jsou považovány za vhodnější alternativu, protože obsahují téměř nezávislé komponenty. DBN-DNN systémy nevyžaduji nekorelovaná data a výsledky rozpoznávání na TIMIT korpusu ukazují, že DBN-DNN, natrénované pomocí příznaků bank filtrů, mají chybovost nižší o 1,7 procent než nejlepší DBN-DNN trénované s MFCC příznaky.

5.1.2 Shrnutí rozdílů mezi DNN a GMM na TIMIT

Hlavním rozdílem jsou podle článku [3] RBM, které slouží jako stavební blok pro předtrénování, jsou instancí tzv. „product of experts“ (součinem hustot rozdělení pravděpodobnosti), kdežto modely mixtur jsou „sum of experts“. Modely součinů byly objeveny ve zpracování řeči teprve nedávno. Modely mixtur s velkým množstvím komponent používají parametry neefektivně, protože každý parametr se aplikuje na zlomek dat, kdežto každý parametr v součinovém modelu se aplikuje na velká množství dat. Přestože jsou DNN a GMM nelineární modely, je jejich druh nelinearity velmi odlišný. DNN nemá problém s modelováním více současných událostí v jednom rámci, protože může používat různé podmnožiny skrytých neuronů k modelování různých událostí. Oproti tomu GMM předpokládá, že každý rámec je generován jednou komponentou mixtury, tudíž nemá efektivní cestu k modelování více současných událostí. Jednou z výhod DNN je, že jsou schopny vytěžit informace z okolních rámců, kdežto GMM mají z více rámců méně užitku, protože potřebují dekorelované vstupy. Nakonec DNN využívají k učení metodu největšího spádu, kdežto GMM používají EM algoritmus nebo jeho rozšíření, které učení GMM dělá lépe paralelizovatelným na clusteru.

Na úspěchy DBN-DNN na TIMIT úloze se navázalo dalšími experimenty a projekty s větším množstvím dat (viz následující podkapitoly). Aby DBN-DNN pracovaly dobře, pro velké slovníky je důležité místo monofonů použít trifony. Ty jsou reprezentovány v HMM kontextově závislými stavy, jejich předpovídání poskytuje několik výhod oproti předpovídání monofonů. Například pro každý rámec poskytují více bitů informací a umožňují použití silnějšího trifonového HMM dekodéru. Použitím kontextově závislých HMM stavů umožňuje neuronové síti o dvou skrytých vrstvách (bez přetrénování) překonat state-of-the-art BMMI (boosted maximum mutual information) natrénované GMM-HMM systémy viz [3]. S použitím více vrstev a přetrénování se může dojít ještě k lepším výsledkům.

5.2 Bing mobile voice search

První úspěšné použití akustických modelů [4] založených na DBN-DNN pro úlohy s velkým slovníkem využívalo data nasbíraná z aplikace Bing mobile voice search (BMVS). Model byl natrénován na 24 hodinovém korpusu. Ten obsahoval mnoho

akustických variací, které byly způsobeny šumem, muzikou, hlasy na pozadí, akcentem, špatnou výslovností, nerozhodností, opakováním, přerušeními a různými typy mobilních telefonů. Výsledky ukázaly, že nejlepší DNN-HMM akustický model natrénovaný s kontextově závislými stavy dosáhl přesnosti (angl. Accuracy) 69,6 % na testovací sadě.

Pro neuronovou síť bylo použito pět generativně předtrénovaných skrytých vrstev, každá s 2048 neurony, a byly natrénovány ke klasifikaci prostředního rámce z 11 rámců na vstupu. Výstupem byly posteriorní pravděpodobnosti 761 kontextově závislých stavů.

Navíc k demonstrování toho, že DBN-DNN můžou poskytnout lepší výsledky na úlohy s velkým slovníkem, bylo prozkoumáno několik dalších problémů. Bylo zjištěno, že trénování sítě pro rozpoznání trifonů, jakožto kontextově závislých stavů HMM, bylo lepší než použití monofonových stavů. Také bylo potvrzeno, že nižší chybovost systému, který byl použit k akustickému zarovnání (angl. forced aligment), způsobuje nižší chybovost výsledného modelu neuronové sítě. Dalším experimentem bylo rozšíření trénovacího korpusu z 24 hodin na 48 hodin. V tomto experimentu se zjistilo, že předtrénování je důležité, protože inicializovalo DBN-DNN váhy k místu, ze kterého bylo trénování efektivnější. Nicméně mírné zvýšení množství neoznačených trénovacích dat nemělo významný vliv na finální rozpoznávací skóre (69,6 % oproti 69,68 %). Pokud bylo ale použito stejné množství předzpracovaných trénovacích dat pro diskriminativní trénování, zlepšil se výkon DNN-HMM systému a přesnost vzrostla na 71,7 %.

5.3 Switchboard

V tomto pokusu [17] byl použit obdobný postup jako v předchozím experimentu BMVS. K natrénování DNN-HMM bylo použito více než 300 hodin trénovacích dat. Topologie neuronové sítě se skládala ze sedmi skrytých vrstev, přičemž každá obsahovala 2048 neuronů. Každé dvě sousední vrstvy byly plně propojeny. Výstupem byly pravděpodobnosti 9304 svázaných stavů. Použití DBN-DNN redukovalo chybovost o 8,9 % oproti GMM-HMM řešení. Také bylo zjištěno, že DNN-HMM systém natrénovaný na 309 hodinách rozpoznává zhruba stejnou přesností jako GMM-HMM systém natrénovaný na 2000 hodinách.

Další experimenty na Switchboard úloze potvrdily, že pozoruhodné snížení chybovosti je způsobeno: 1) přímým modelováním svázaných trifonových stavů použitím DBN-DNN; 2) efektivním využitím sousedních rámců; 3) silným modelovacím možnostem při použití hlubších sítí. Generativní předtrénování DBN-DNN vedlo k nejlepším výsledkům a redukovalo chybovost o méně jak procento a při použití pěti a více vrstev byla redukce chybovosti ještě menší. Pro jazyky s nedostatečným množstvím zdrojů předpřipravených trénovacích dat by mělo být předtrénování užitečnější.

Další experimenty naznačují, že techniky úprav příznaků jako HLDA a VTLN, které jsou běžně používány v GMM-HMM systémech, jsou více užitečné pro mělké neuronové sítě než pro DBN-DNN. Jako eventuální příčina takového chování byla uvedena možnost neuronové sítě se naučit obdobné příznaky v nižších vrstvách sítě.

5.4 Google voice input

Google Voice Input přepisuje hlasové požadavky na: vyhledávání, krátké zprávy, emaily a akce uživatele v mobilním přístroji. Lze to tedy označit za úlohu [18] s velkým slovníkem, která používá jazykový model navržený pro směs vyhledávacích požadavků a diktování.

Plnohodnotný model Googlu pro tuto úlohu byl postaven z velkého korpusu. Používal GMM-HMM model složený z kontextově závislých trifonů. Topologie HMM byla třístavová zleva doprava. Tento model měl 7969 senonových stavů a používal PLP příznaky, které byly získány lineární diskriminativní analýzou (LDA) z akustického vstupu. GMM-HMM model byl použit k získání 5870 hodin zarovnaných trénovacích dat pro DBN-DNN akustický model, který předpovídá posteriorní pravděpodobnosti HMM stavů. DBN-DNN obsahovala 4 vrstvy s 2560 neurony. Sousední vrstvy byly plně propojené. Jako vstup sloužilo 11 sousedních framů, každý o 40 příznacích logaritmických bank filtrů. Každá z DBN-DNN vrstev byla předtrénována jednou epochou jako RBM a potom výsledná DNN byla diskriminativně trénována pro jednu epochu. Váhy s magnitudami pod prahem byly permanentně nastaveny na 0 s každou čtvrtinou epochy trénování. Jedna třetina vah ve finální síti obsahovala pouze nuly. Na testovací sadě anonymní řeči byla chybovost

(WER=100-accuracy) 12,3 % a redukovala tak relativně o 23 % chybovost nejlepšího GMM systému pro tuto úlohu.

5.5 YouTube

V této úloze [18] bylo cílem přepsat mluvený text ve videích na portálu YouTube na psaný text. Narozdíl od předchozího projektu Google, tato aplikace neměla silný jazykový model, který by sloužil ke zkrocení interpretace akustických informací. Proto bylo důležité vytvořit přesný akustický model. Trénovací korpus obsahoval 1400 hodin akusticky zarovnaných dat. Vstupem bylo 9 rámců MFCC příznaků, které byly transformovány pomocí LDA. Výstupem bylo 17552 trifonových stavů. Navrhovaná DNN byla složena ze 4 skrytých vrstev. V první skryté vrstvě se nacházelo 2000 neuronů a ve všech ostatních 1000 neuronů. Bylo provedeno zhruba 10 epoch trénování (pravděpodobně předčasně zastaveno) a DBN-DNN se zlepšilo v chybovosti o 4,7 % oproti baseline systému s WER 52,3 %. (Pozn. chybovost rozpoznávání je vyšší bez použití jazykového modelu.)

5.6 Zpravodajství

DNN byly také úspěšně aplikovány na úlohu rozpoznání řeči vysílání zpráv v AJ z let 1996 až 1997 [3]. K zarovnání akustických dat byl použit baseline systém. Trénovací korpus obsahoval 50 hodin dat. DBN-DNN bylo natrénováno s příznaky SAT+DT. Topologie byla složena z 6 skrytých vrstev a každá obsahovala 1024 neuronů. Na vstup přicházelo 9 rámců a výstupní vrstva byla složena z 2220 neuronů. Nejprve bylo provedeno generativní předtrénování pomocí RBM. Poté následovala trénovací fáze, která byla rozdělena do 2 částí. Během první části se přizpůsoboval parametr učení a byla stanovena výchozí hodnota vah pro druhou část, která představovala samotné trénování sítě. DNN-HMM takto zlepšila chybovost oproti GMM-HMM systému o 1,3 % (snížení WER z 18,8 % na 17,5 %).

5.7 Shrnutí

Pro účely práce byly zajímavé topologie sítí, které se pohybovaly od 4 do 8 skrytých vrstev a od 1000 do 2500 neuronů ve skrytých vrstvách. Do sítě vstupovaly příznaky z 9 až 11 rámců. Práci bylo možné porovnat vzhledem k podobným velikostem trénovacích korpusů a podobnými testovacími sadami s pokusem uvedeným v kapitole 5.6.

6 Experimenty

V této části budou představeny nástroje použité pro trénování neuronových sítí a základní nastavení parametrů. V dalších podkapitolách budou popsány tři pokusy. První dva se zabývaly topologií a předtrénováním neuronové sítě a v posledním byl pozorován vliv množství dat v trénovacím korpusu na přesnost rozpoznávání.

6.1 Popis toolkitu

K trénování byly použity modifikované skripty z knihovny Theano [15] pro Python, která umožňuje definovat, optimalizovat a ohodnotit matematické výrazy včetně efektivního použití multidimenzionálních polí. Kombinuje pohodlí syntaxe NumPy s vyšší rychlostí nižších jazyků. Před spuštěním skriptu Theano automaticky zkompiluje vybrané matematické výrazy do jazyka C++ nebo v případě GPU do jazyka CUDA a dynamicky je umístí k načteným modulům. Algoritmy strojového učení jsou rychlejší až 7,5 krát než konkurenční řešení (včetně těch implementovaných v C++). Rychlost GPU je oproti CPU rychlejší až 44 krát. V příloze J byl uveden postup pro instalaci knihovny Theano.

K trénování byly použity modifikované scripty z dokumentace knihovny [16]. K jejich úpravě bylo použito vývojové prostředí Aptana Studio 3, ale vzhledem k pomalejšímu debugování by lepší volbou bylo Visual Studio 2010 (u novějších verzích mohou být problémy s kompatibilitou). Získané hodnoty přesnosti a rozpoznání testovacích sad provedl program NanoDictateT.

6.2 Základní nastavení

Pro všechny experimenty byly použity tyto konstanty:

$\eta = 0{,}08$ $(učící\ parametr)$,
$\alpha = 1$ $(momentum)$,
$i = 429$ $(počet\ vstupních\ neuronů\ resp.\ počet\ příznaků)$,
$o = 3180$ $(počet\ výstupních\ neuronů\ resp.\ stavů\ HMM)$.

Příznaky byly vytvořeny z 11 rámců akustických dat a každému rámci odpovídalo 39 MFCC příznaků. Neuronová síť byla trénována pro rozpoznání „prostředního“ rámce.

Pro trénování byl k dispozicí trénovací korpus s 56 hodinami polské řeči. Vyhodnocení přesnosti vytvořených modelů neuronových sítí bylo posouzeno na 3 testovacích sadách:

- Broadcast – obsahuje věty z oblasti vysílání televizních a rozhlasových zpráv,
- Justice – obsahuje věty v právním kontextu,
- Pomocne – obsahuje samostatná slova.

Přesnost současně používané metody (angl. baseline), která vychází z GMM-HMM systému, byla vyhodnocena pro jednotlivé testovací sady v tabulce 1.

Přesnost baseline modelů [%]		
Broadcast	Justice	Pomocne
73,14	88,09	85,16

Tabulka 1

Pro trénování byly k dispozici dvě pracovní stanice, dále označeny jako P1, P2 s grafickými kartami NVIDIA GeForce GTX 780. P1 disponovala procesorem Intel(R) Core(TM) i7-4770 3,40GHz. Stanice P2 disponovala procesorem Intel(R) Core(TM) i7-3770K 3,50GHz (zdroje: Systém a Správce zařízení ve Windows 7 Professional) a byla využívána zejména během druhé poloviny prosince a začátkem ledna.

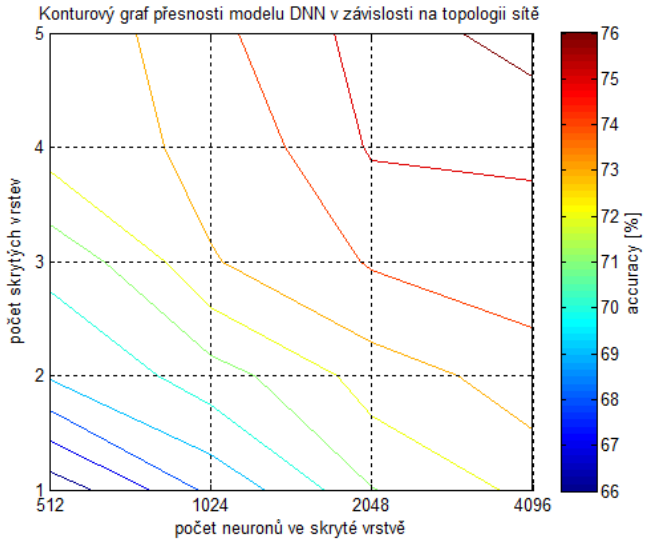
V kapitole byl počtem vrstev myšlen počet skrytých vrstev v neuronové sítí (vstupní a výstupní vrstva nebyla tedy brána v úvahu) a počet neuronů v každé ze skrytých vrstev bylo vyjádřeno slovy šířka neuronové sítě. Pro účely této práce měly všechny skryté vrstvy stejnou šířku.

6.3 Optimální topologie sítě

Cílem tohoto experimentu bylo zjistit optimální topologii sítě, resp. optimální počet skrytých vrstev a neuronů v plně propojených neuronových sítích (každé dvě sousední vrstvy neuronů tvoří bipartitní graf). Na základě rešerše (viz kapitola 5) se rozhodlo otestovat šířky o 512, 1024, 2048 a 4096 neuronech. Experimenty byly prováděny pro 1 až 5 skrytých vrstev.

6.3.1 Hodnocení podle přesnosti

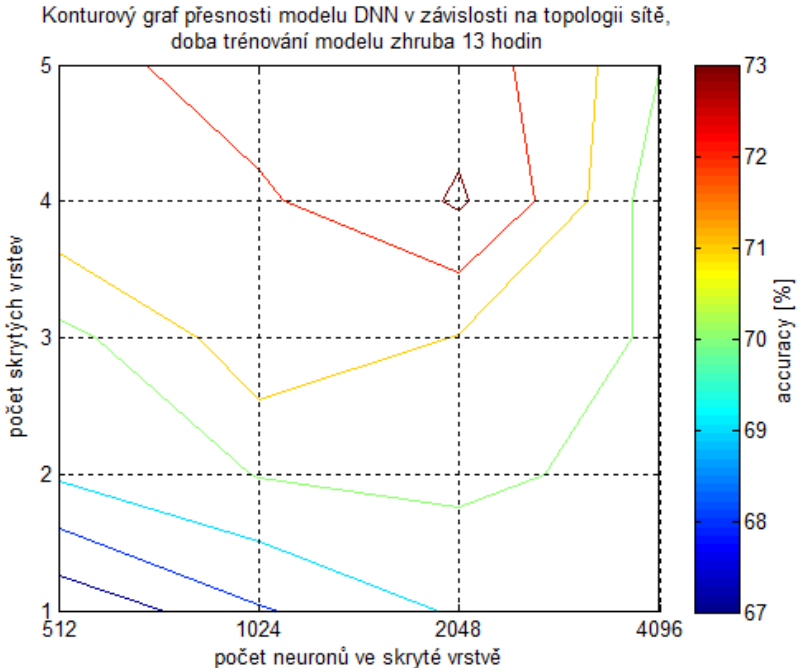
Na grafu 6.1 byla znázorněna přesnost modelu, jehož trénování bylo zastaveno po 80 epochách. (Počet epoch vyjadřuje počet průchodů trénovacích dat celou sítí.) Z výsledků bylo pozorováno, že natrénování jednovrstevné sítě široké 4096 neuronů a pětivrstevné sítě široké 1024 neuronů mělo zhruba stejnou přesnost rozpoznávání. Z výsledků také vyplynulo, že hlubší sítě se stejnou šířkou byly schopny se naučit více z těch samých dat při stejném počtu průchodů, což víceméně bylo očekávané, neboť obsahovaly mnohem více „paměťových jednotek“ (vah), které byly trénovány. Při větším počtu vah ovšem trvá trénování déle. Například pro síť o dvou skrytých vrstvách a 4096 neuronech na vrstvu trvalo trénování 44 hodin, kdežto u pětivrstevné sítě o 1024 neuronech na skrytou vrstvu, které má podle grafu 6.1 obdobnou přesnost, trvalo trénování 23 hodin.



Graf 6.1

Proto pro některé případy, např. při nedostatečné výpočetní kapacitě, by se mohlo pro posouzení optimality topologie sítě vzít v úvahu i doba trénování (viz graf 6.2). Při takto nastavených podmínkách vyplynulo, že hluboké sítě opět byly přesnější než mělké, ale propad v přesnosti byl také zaznamenán u sítí širokých 4096 neuronů. Další grafy pro 23 a 35 hodin byly umístěny do přílohy B, neboť jejich výsledky korespondovaly s grafem 6.2, přičemž se zdálo, že s přibývající dobou trénování se oblast „optimálních“ topologií sítí zvětšuje směrem k širším sítím a hlubším sítím. Při

35 hodinách trénování překonala pětivrstevná síť o šířce 2048 neuronů čtyřvrstevnou o stejné šířce.

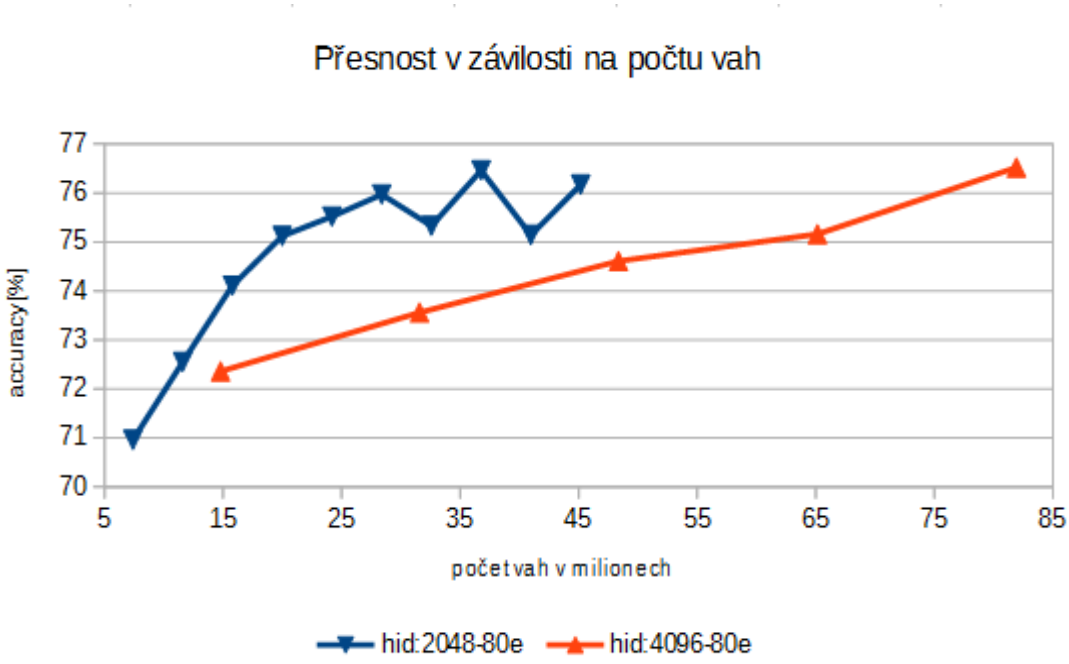


Graf 6.2

V kapitole 4 bylo uvedeno, že síla neuronových sítí spočívá ve schopnosti si zapamatovat různé vztahy mezi vstupy do jednotlivých neuronů, což obstarávají hodnoty vah. Při experimentech s topologií sítí nastala situace, kdy rozdílné modely měly přibližně stejný počet vah. Teoreticky by to mohlo znamenat obdobný potenciál, ale ze získaných výsledků bylo zjištěno, že hlubší sítě dosahují lepších výsledků než sítě širší viz graf 6.3, ve kterém byly zobrazeny přesnosti sítí širokých 2048 a 4096 neuronů v závislost na počtu vah. Počet vrstev jednotlivých sítí lze zjistit počtem bodů zleva na jednotlivých úsečkách, resp. bylo natrénováno 10 vrstev pro síť širokou 2048 neuronů a 5 vrstev pro síť širokou 4096 neuronů. Trénování modelů bylo zastaveno po uplynutí 80 epoch. Počty vah byly vypočítány podle rovnice 19, ve které byly brány v potaz i prahové hodnoty (angl. bias term).

$$N=(I+1)S+(S+1)S(V-1)+(S+1)O \qquad (19)$$
$$N=S^2(V-1)+S(I+V+O)+O,$$

kde N je celkový počet vah, I je počet neuronů ve vstupní vrstvě, S počet neuronů ve skryté vrstvě, V počet skrytých vrstev, O počet neuronů ve výstupní vrstvě.

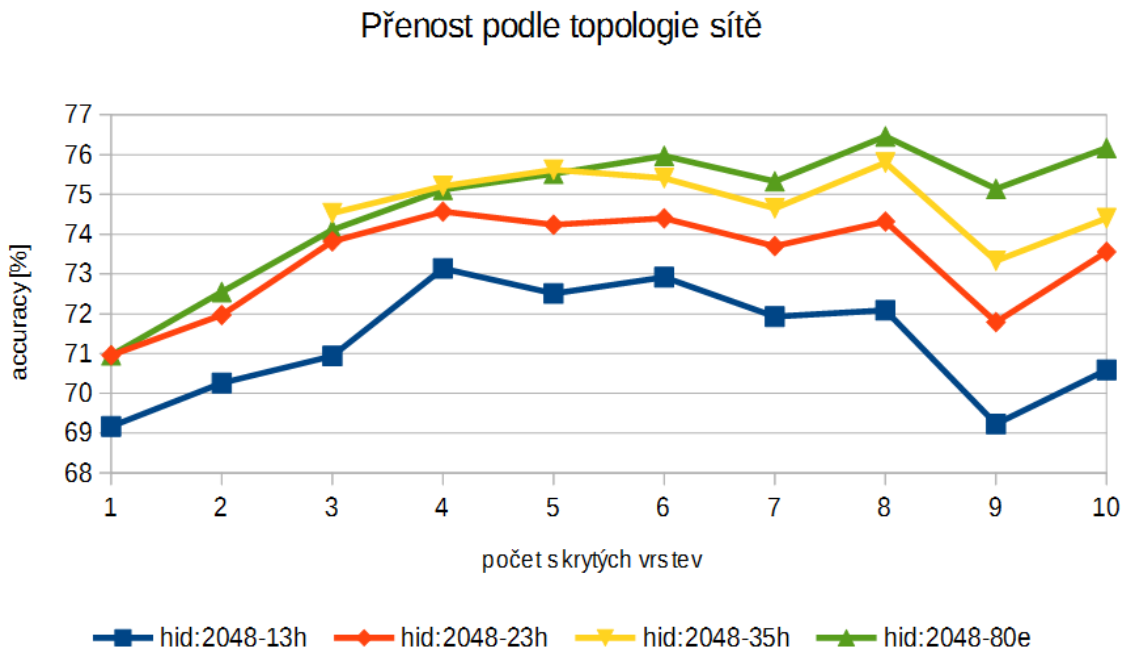


Graf 6.3

Další výsledky pro sítě široké 512 a 1024 neuronů byly umístěny v příloze D. Na základě grafu 6.3 bylo také usouzeno, že v případě použití hlubších sítí lze dosáhnout obdobné přesnosti i s polovičním počtem vah. Úskalím, které nebylo pozorováno, ale bylo zmíněno v článku [11], by mohl být konstantní učící parametr pro všechny vrstvy sítě, neboť jsou pak do nižších vrstev zpětně šířeny nižší hodnoty.

Ve článcích [4], [8], [12] bylo zmíněno, že hlubší sítě se zdají být výhodnější nežli sítě s nižším počtem vrstev, což se zdálo být potvrzeno grafem 6.2 a přílohou B. Proto bylo vytvořeno dalších 5 modelů neuronových sítí o šířce 2048 neuronů a počtem vrstev 6 až 10. Přesnosti všech 10 modelů neuronových sítí o šířce 2048 neuronů byly znázorněny v grafu 6.4 ve čtyřech variantách. První tři byly předčasně zastaveny v určitý čas (ve 13, 23 a 35 hodin) a čtvrtá, zelená úsečka v grafu 6.4, byla zastavena po uplynutí 80 epoch. Výhoda hlubších sítí se neprojevila, vyjma osmivrstevné sítě, což bylo považováno za výjimku při konvergenci k lepšímu lokálnímu minimu. Hlubší sítě předvedly lepší výsledky, pokud bylo bráno v potaz zastavení po uplynutí určitého počtu epoch, ale v takové posuzování, jak bylo již dříve zmíněno, by se mohlo zdát nespravedlivé, neboť trénování desetivrstevné sítě po 80 epoch trvalo zhruba 65 hodin,

kdežto u pětivrstevné doba činila 37 hodin. S rostoucí dobou trénování bylo pozorováno, že pětivrstevná síť překonala čtyřvrstevnou a mohlo by se zdát, že tento trend může pokračovat i pro další vrstvy, neboť síť o větším počtu vrstev by měla teoreticky mít alespoň stejné modelovací schopnosti jako stejně široká síť o nižším počtu vrstev. Z toho vyplývá nezodpovězená otázka, jestli současný princip trénování toto umožňuje nebo jestli je limitován.

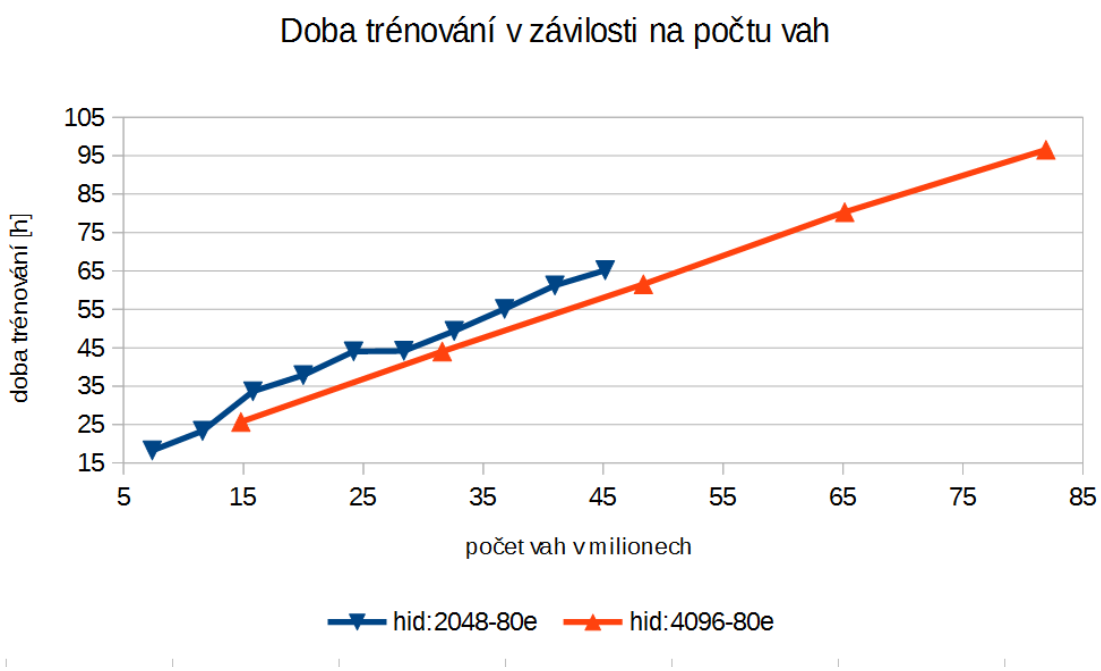


Graf 6.4

Z dalších testů vyplynulo, že síť o pětivrstvách a 1024 neuronech na skrytou vrstvu překonává přesnost osmivrstevné sítě s 2048 neurony na vrstvu (trénované po 81 epoch), po uplynutí 166 epoch s časovou rezervou zhruba 8 hodin.

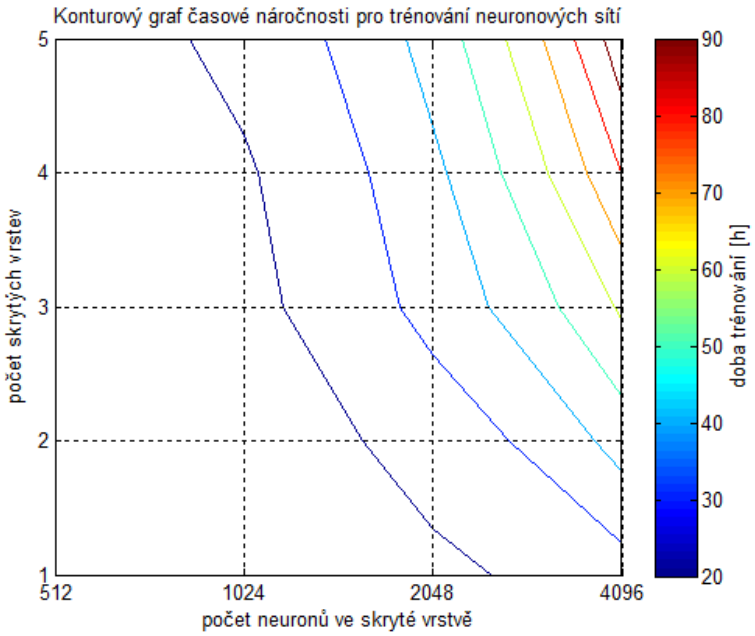
6.4 Hodnocení podle doby trvání

Vzhledem k nelinearitě výpočtu vah a k tomu, že váhy jsou tou hlavní jednotkou, která ovlivňuje chování sítě a jsou trénovány algoritmem, který se považuje za tzv. bottleneck při trénování byl vytvořen graf 6.5, ze kterého se lze domnívat, že náročnost trénování sítě stoupá zhruba lineárně. Obdobný graf s pro šířky skrytých vrstev 512 a 1024 neuronů lze nalézt v příloze E, vzhledem ke kratší době trénování jejich hodnoty mohou být více ovlivněny faktory popsanými v podkapitole 6.5.1. Obdobně jako u předešlého grafu počet bodů počítaný zleva udává počet skrytých vrstev pro vybraný počet vah. Z grafu vyplývá, že zmiňovaná nelinearita trénování širších a hlubších sítí je spíše dána počtem jejich vah a také, že hlubší síť trvá déle natrénovat při stejném počtu epoch, pravděpodobně kvůli tomu, že pro každou další vrstvu je nutné vypočítat nové hodnoty chyby, viz kapitola 2.3.



Graf 6.5

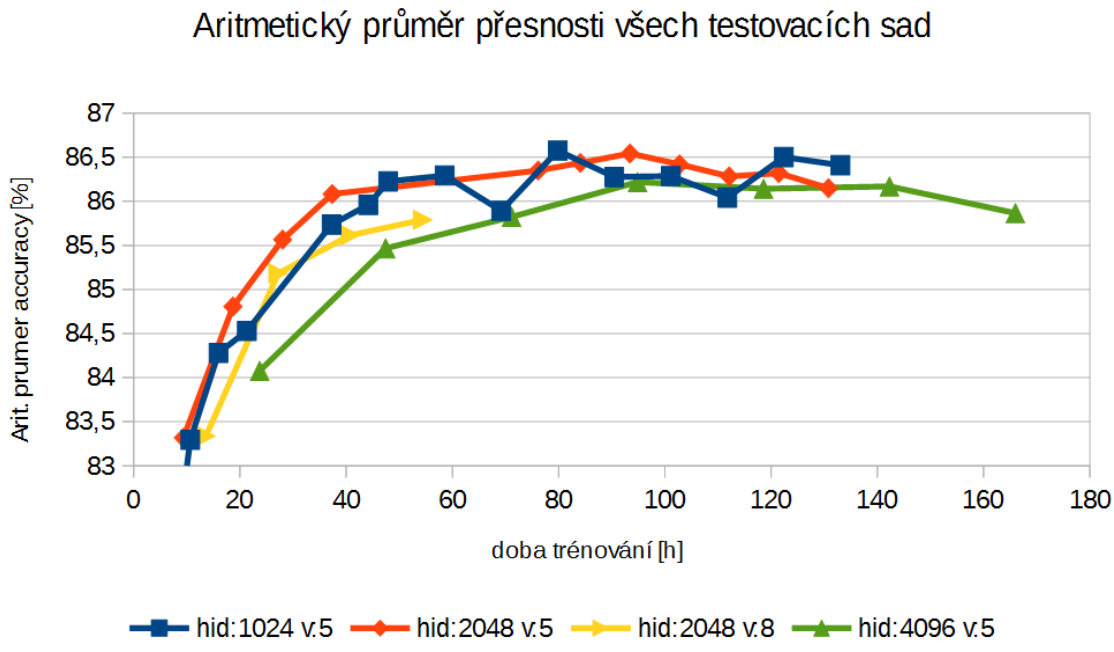
Tento graf vyjadřuje, které sítě lze natrénovat za stejnou dobu (např. dvouvrstevná síť s 2048 neurony ve skryté vrstvě vyžaduje přibližně stejnou dobu jako pětivrstevná síť o šířce 1024 neuronů). Jak vyplývá z předchozí kapitoly 6.3.1 a přílohy B, se zdá lepší trénovat sítě hlubší.



Graf 6.6

6.5 Nejlepší model

Na základě získaných znalostí se přistoupilo k dotrénování 5 vrstevných síťí o šířkách 1024 a 2048 neuronů. Oba modely byly trénovány po několik dní a jejich možnosti byly vyjádřeny grafem 6.7, ve kterém byly navíc vyobrazeny dvě další sítě – pětivrstevná o šířce 4096 neuronů a 8 vrstevná o šířce 2048, pro kterou bylo méně dat. Počet epoch nutný k natrénování jednotlivých modelů byl (podle barev v grafu): modrá 500, červená 280, zelená 140 a žlutá 80 epoch. Sítě o šířce 1024 a 2048 neuronů si byly vzhledem k aritmetickému průměru přesnosti vyrovnané. Nebylo shledáno, že by jedna ze sítí měla rychlejší dobu konvergence, proto konečný verdikt je odvozen nejen na základě 4 setin procenta lepšího aritmetického průměru pro síť širokou 1024 neuronů, ale i na základě toho, že bylo přihlédnuto k možné nižší době rozpoznávání v případě menší sítě.



Graf 6.7

6.5.1 Další faktory

Některé z testů měly odstup několika měsíců, nebo se konaly na různých pracovních stanicích. Z těchto důvodů mohlo dojít v případě dotrénování či porovnání dat z různých pracovních stanic k menším nepřesnostem (např. různým rozdělením trénovacího korpusu) viz kapitola 2.4 o krosvalidaci. V případě stejné pracovní stanice

a dotrénování epoch s jiným rozdělením trénovacího korpusu mohlo dojít dle vědeckého článku [11] k ovlivnění přesnosti klasifikace. V případě rozdílných trénovacích stanic se mohlo jednat o konvergenci k různým lokálním optimům. Rozdělení jednotlivých testů podle příslušných pracovních stanic bylo uvedeno v příloze C.

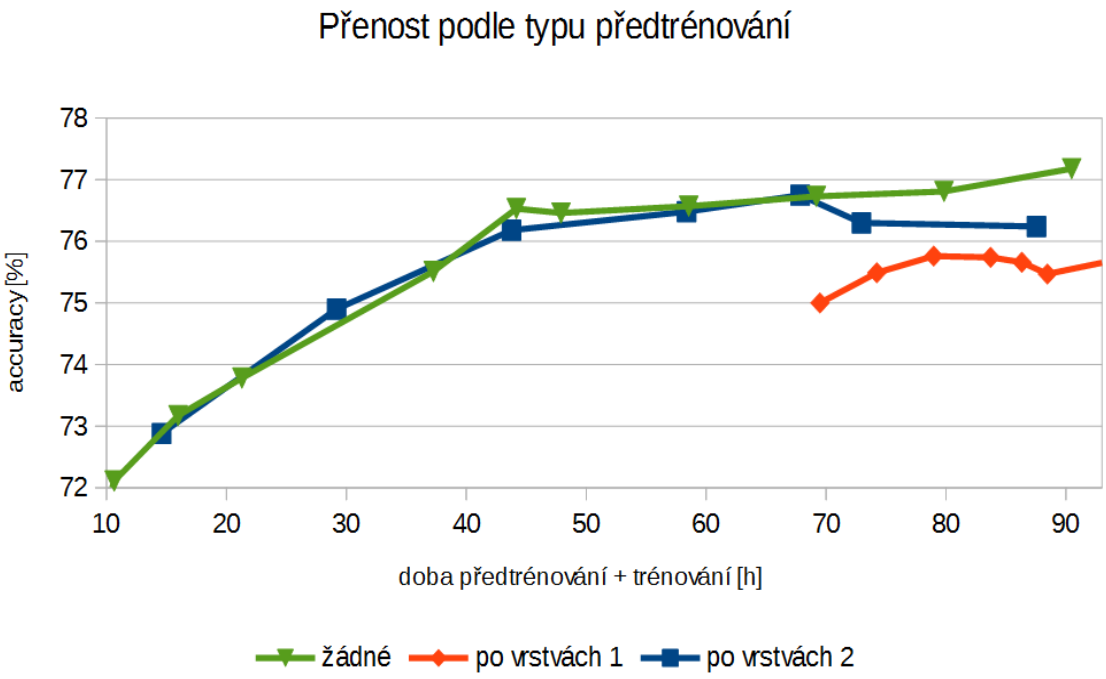
Mezi další faktory, které mohly ovlivnit výsledky doby trénování sítě, patří spouštění procesu rozpoznávání během trénování sítí, dále možné aktualizace systému a jiné naplánované úlohy, trochu rozdílné pracovní stanice a sdílení P2 s dalšími uživateli, kteří využívali zejména nevyužité periferie (Modelování sítě probíhá zejména na grafické kartě.).

V tomto experimentu byla využita i pracovní stanice č. 2, na které byly natrénovány modely pro tyto topologie sítí: 1, 3 a 5 vrstev o šířce 512 neuronů; 4,5 vrstev o šířce 1024 neuronů a 3, 5, 7, 9 vrstev o šířce 2048 neuronů. Na základě výsledků, kde dochází pro tyto případy k výkyvům, bylo zjištěno, že bylo použito jiné rozdělení trénovacího korpusu na trénovací, validační a testovací data. Zároveň tato stanice byla sdílená, a proto dle náporu mohlo dojít k menším výkyvům v hodnotách dob trénování. Z těchto důvodů byly výsledky odděleny.

6.6 Předtrénování

Cílem této kapitoly je zjistit vliv předtrénování popsaného v kapitole 4.2, kdy byla síť diskriminativně trénována vrstvu po vrstvě. Oproti popsané metodě byl rozdíl v tom, že nebylo zablokováno trénování vah předchozích vrstev, což mohlo mít jak pozitivní tak negativní důsledky. Jako pozitivum šlo chápat možnost dalšího dotrénování vah a jako zápor nemožnost vyskočení z dříve nalezeného lokálního minima. Toto předtrénování proběhlo dvěma způsoby: 1) každá vrstva byla natrénována co nejlépe a 2) po každých 5 epochách byla přidána nová vrstva, dokud nebylo dosaženo požadovaného množství vrstev. Každá vrstva v prvním typu předtrénování prošla alespoň 100 epochami, přičemž vyhodnocení bylo provedeno na základě výsledku hodnocení klasifikace pro testovací část trénovacího korpusu. Takto předtrénované sítě poté byly diskriminativně dotrénovány. Topologie sítě byla zvolena s přihlédnutím na možnou delší dobu trénování pěti vrstev o šířce 1024 neuronů.

Podle grafu 6.8 a výsledků v příloze H se zdálo být diskriminativní předtrénování zbytečné. Dokonce se pro první ze jmenovaných způsobů, kdy je každá vrstva



Graf 6.8

natrénovaná s nejlepším možným výsledkem na testovací části trénovacího korpusu, zdá být předtrénování kontraproduktivní, což může být způsobeno dlouhou dobou trénování jednotlivých vrstev.

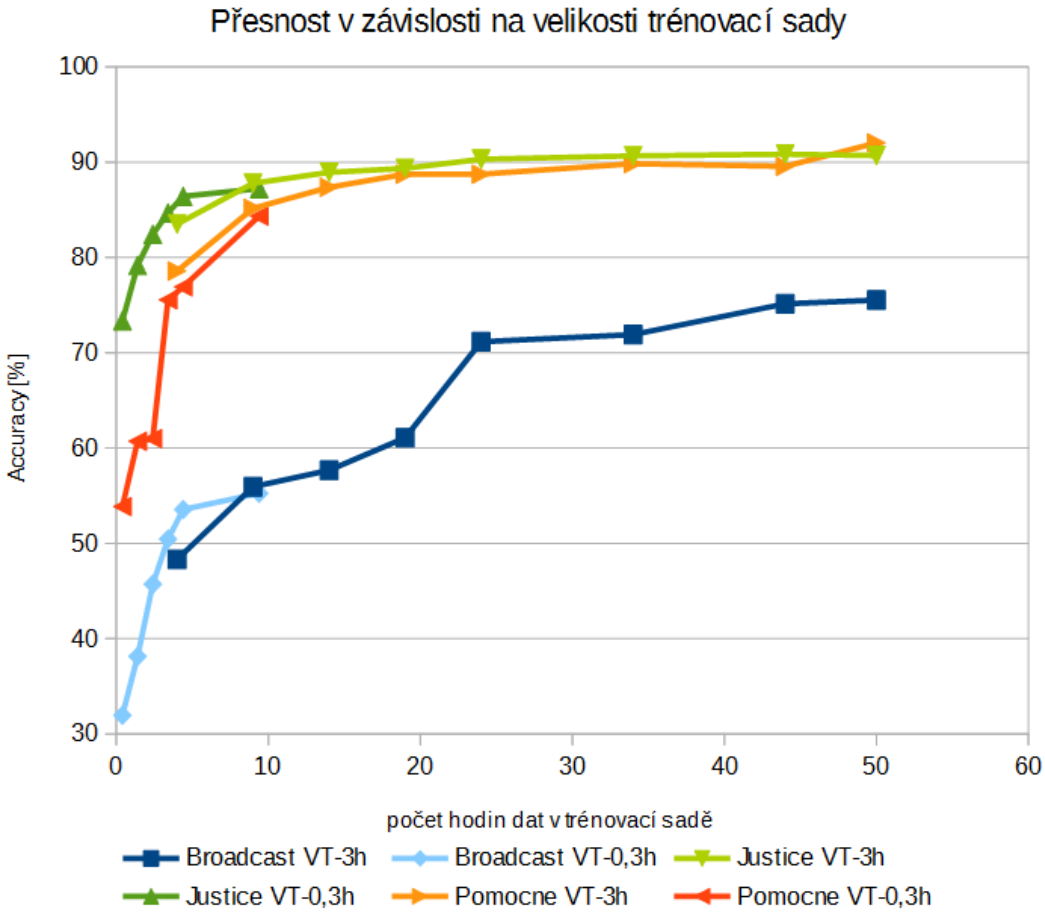
6.7 Množství dat

Cílem této kapitoly bylo zjistit kolik dat je zapotřebí k natrénování dobrého modelu DNN. Resp. pokud bylo přidáno více dat, o kolik se přesnost rozpoznávání sítě zlepšila. Topologie sítě byla nastavena na 5 vrstev o šířce 2048 neuronů.

V kapitole 2.4 bylo uvedeno, že trénovací korpus se rozděluje z určitých důvodů do 3 částí: trénovací sada, validační sada a testovací sada dat. Symboly VT byly v následujících grafech označeny délky validační a testovací sady. Pro více dat byly použity validační sady o délce 3 hodin, to samé pro testovací sadu a pro méně dat byly použity validační a testovací sady o délce 18 minut. Pro každý korpus byly vygenerovány nové trénovací, validační a testovací sady.

U testovacích korpusů Justice a Pomocne se zdálo z grafu 6.9 patrné, že přínos nových dat nad 25 až 30 hodin byl malý a celkový průběh jejich křivek připomínal

logaritmický charakter, ale u testovací sady Broadcast dochází v některých případech ke skokovým zlepšením (např. u zvětšení testovací sady z 34 hodin na 40 hodin došlo ke zlepšení o více jak 3 procenta). Možné důvody pro toto chování můžou být následující: testovací sadu Broadcast je těžší rozpoznat, tudíž je větší prostor pro zlepšení, nebo neznáme závislosti v trénovacím korpusu.



Graf 6.9

Také bylo zjištěno, že doba trénování vzhledem k množství dat stoupá lineárně (viz příloha G).

6.8 Shrnutí

Dalo by se říci, že čas nutný pro trénování stoupá lineárně s počtem vah a že modelování hlubších sítí o stejném počtu vah trvalo nepatrně déle (o jednotky hodin) oproti sítím širším. Delší doba trénování byla v těchto případech ohodnocena vyšší přesností při rozpoznávání. Při srovnání jednotlivých topologií sítí, jejichž

trénování bylo předčasně zastaveno po uplynutí stejného časového úseku, bylo zjištěno, že sítě o 1024 a 2048 neuronech ve skryté vrstvě a o 4 a 5 skrytých vrstvách se zdají být nejvhodnější. Navrhlo se tedy trénovat 5 vrstevné sítě o 1024 a 2048 neuronech o několik desítek hodin déle, aby se zjistila lepší topologie. Při vyhodnocení na testovací sadě Broadcast se zdála mít užší síť o něco lepší výsledky a při posouzení výkonu na ostatních testovacích sadách rovněž. Vybraná topologie dosáhla zlepšení v absolutních hodnotách oproti baseline systému u testovací sady Broadcast o 3,67 %, u testovací sady Justice o 2,8 % a u testovací sady Pomocne o 6,87 %.

Druhým pokusem byl zkoumán vliv na přesnost rozpoznávání při použití diskriminativního předtrénování neuronové sítě. Zjistilo se, že předtrénování nemělo žádný vliv a v některých případech dokonce vliv negativní.

Další experiment se zabýval vlivem množství dat na přesnost vytvořených modelů neuronové sítě. Bylo zjištěno, že u testovacích sad Pomocne a Justice byl vliv nových dat nad 30 hodin malý, kdežto u testovací sady Broadcast docházelo opakovaně ke skokovým zlepšením. Z toho bylo usouzeno, že množství dat závisí na kontextu, ve kterém bude rozpoznávač používán, resp. prostor pro zlepšení byl u testovacích sad Pomocne a Justice menší než u testovací sady Broadcast.

Celkem bylo natrénováno 45 různých neuronových sítí, které se lišily v množství dat, typu předtrénování nebo v topologii. Celková výpočetní doba trénování bez otestování modelů činila 62 dní. Vzhledem k tomu, že trénování probíhalo zejména na GPU, mohlo testování probíhat souběžně na CPU, a tím se ušetřil čas. Testování bylo výrazně kratší a pohybovalo se řádově v desítkách minut pro vybraný model. Většinou bylo vybráno 4 až 7 modelů pro jednu topologii. Modely byly vybírány na základě nejlepšího skóre na testovací části trénovacího korpusu nebo na základě počtu uplynutých epoch ve výjimečných případech také na základě uplynuté doby. Časové údaje byly vypočítány na základě logů z trénování, viz příloha CD, a nebo byly vypočítány z průměrné doby trvání jedné epochy, neboť téměř všechny epochy trvají stejně. Výjimkou bylo pouze několik prvních epoch po inicializaci. Veškeré výsledky testů jsou přiloženy na CD.

7 Závěr

V práci byla popsána základní problematika a provedena rešerše LVCSR projektů zabývajících se akustickým modelováním pomocí neuronových sítí. Návrhy, výsledky a postupy získané z rešerše inspirovaly experimenty v praktické části práce. Tato rešerše byla doplněna o současné novinky v systémech rozpoznávání řeči.

Na dvou pracovních stanicích ústavu ITE byl zprovozněn software umožňující trénování neuronových sítí na GPU. Potom proběhlo trénování modelů neuronových sítí na 56 hodinách spojité polské řeči. Po vyhodnocení modelů na testovacích sadách vyplynulo, že jako nejlepší se zdá topologie o šířce 1024 neuronů a pěti skrytých vrstvách. Na testovacích sadách došlo k absolutnímu zlepšení přesnosti o 3 až 7 % oproti současně používanému systému. Obdobná topologie byla použita i v projektu popisovaném v podkapitole 5.6, kde došlo ke zlepšení pouze o 1,8 %. Dále byla vyzkoušena metoda diskriminativního předtrénování, která se ukázala zbytečná, neboť diskriminativní trénování modelu po stejnou dobu vedlo ke stejným, nebo lepším výsledkům.

Poslední část experimentů se zabývala množstvím dat nutných pro trénování, které by mělo být nižší, než u současných modelů. Ukázalo se, že tento experiment byl silně závislý na zvolené testovací sadě. U dvou případů bylo dostačujících 30 hodin trénovacích dat. Přidáváním dalších dat totiž docházelo pouze k zanedbatelným zlepšením řádově o desetiny procent, kterých by ale pravděpodobně šlo dosáhnout i delší dobou trénování. V případě testovací sady obsahující řeč z televizního zpravodajství se zdálo, že k dalším výrazným zlepšením by mohlo docházet i po přidání více dat, než bylo pro experiment k dispozici.

Vzhledem k množství nastavitelných parametrů neuronových sítí bylo hlavním přínosem práce vytvoření částečného přehledu výhodnosti některých voleb. Přestože by výsledné modely zlepšily výsledky současných systémů, objevují se nové metody trénování, nové druhy aktivačních funkcí a také další postupy, které se zdají být nadějné. Výsledky práce by měly být použity pro zjištění přínosu nových postupů s navrženými parametry v této práci. Vytvořené modely lze také použít k vylepšení současných systémů rozpoznávání řeči. K jejich spolehlivosti přispějí i výhody neuronových sítí při rozpoznávání zašuměných nebo poškozených pásek. Výsledný

model lze mimo jiné použít i jako iniciální stav pro trénování rozpoznání řeči jiného jazyka. Neuronové sítě obecně můžou uspět u úloh, kde většina ostatních metod selhává. Mezi tyto úlohy patří: rozpoznávání zašumělých nebo poškozených pásek, vytvoření modelů s nedostatkem trénovacích dat nebo rozpoznávání řeči vícero jazyků zároveň.

V dalších experimentech by mohly být odzkoušeny jiné aktivační funkce, dropout přístup k trénování, upřesnění parametru učení pro jednotlivé vrstvy, či dokonce ReLU neurony. V předtrénování bylo zmíněno trénování po jednotlivých vrstvách, ale v žádném článku nebyla nalezena zmínka o přidávání neuronů do šířky. Taková metoda by mohla spočívat v natrénování co nejlepšího užšího modelu, pro který by byla následně změněna hodnota parametru učení, než pro nově přidané neurony resp. váhy. Nebyl také odzkoušen přístup, kdy neuronová sít slouží k vytvoření nových příznaků, angl. tandem approach.

Literatura

[1] KRÖSE, Ben a Patrick van der SMAGT. *An Introduction to Neural Networks*. 8. vyd. University of Amsterdam, 1996. Dostupné z: http://lia.univ-avignon.fr/chercheurs/torres/livres/book-neuro-intro.pdf

[2] RUSSELL, Stuart J. a Peter NORVIG. *Artificial intelligence: a modern approach*. Englewood Cliffs, N.J.: Prentice Hall, c1995, xxviii, 932 p. ISBN 01-310-3805-2.

[3] HINTON, Geoffrey, Li DENG, Dong YU, George DAHL, Abdel-rahman MOHAMED, Navdeep JAITLY, Andrew SENIOR, Vincent VANHOUCKE, Patrick NGUYEN, Tara SAINATH a Brian KINGSBURY. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine*. 2012, s. 82-97. Dostupné z: https://www.cs.toronto.edu/~hinton/absps/DNN-2012-proof.pdf

[4] DAHL, George, Dong YU, Li DENG a Alex ACERO. Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition. *IEEE Trans. Speech and Audio Proc.* 2012, č. 1, s. 30-42. Dostupné z: http://www.cs.toronto.edu/~gdahl/papers/DBN4LVCSR-TransASLP.pdf

[5] NOUZA, J. (ed.) ; Z. KOLDOVSKÝ (ed.), a R. VÍCH (ed.). *ŘEČ A POČITAČ Principy hlasové komunikace,úlohy, metody a aplikace*. 1. Liberec : Technická univerzita, 2009. 235 s. ISBN 978-80-7372-548-8. 1.

[6] BISHOP, Christopher M. *Pattern recognition and machine learning*. New York: Springer Science Business Media, 2006, xx, 738 s. ISBN 03-873-1073-8.

[7] HAYKIN, Simon. *Neural networks and learning machines*. 3rd ed. Upper Saddle River, N.J.: Pearson, c2009, 934 s. ISBN 01-312-9376-1.

[8] MOHAMED, A., G. E. DAHL a G. E. HINTON. Acoustic Modeling using Deep Belief Networks. *IEEE Trans. on Audio, Speech, and Language Processing*. 2012,č. 20, s 14-22. Dostupné z: https://www.cs.toronto.edu/~hinton/absps/speechDBN_jrnl.pdf

[9] SILOVSKÝ, Jan. *Rozpoznávání mluvčích v záznamech televizních a rozhlasových pořadů*. Liberec, 2006. Diplomová práce. Technická univerzita v Liberci. Vedoucí práce prof. Ing. Jan Nouza, CSc.

[10] ČERVA, Petr. *Řízená a neřízená adaptace na mluvčího v systémech rozpoznávání řeči*. Liberec, 2007. Disertační práce. Technická univerzita v Liberci. Školitel prof. Ing. Jan Nouza, CSc.

[11] DENG, L., G. E HINTON a B. KINGSBURY. *New types of deep neural network learning for speech recognition and related applications: An overview* In IEEE International Conference on Acoustic Speech and Signal Processing (ICASSP 2013) Vancouver, 2013. Dostupné z: https://www.cs.toronto.edu/~hinton/absps/overview13.pdf

[12] MOHAMED, A., G. E. HINTON, a G. PENN. *Understanding how Deep Belief Networks perform acoustic modelling* In IEEE International Conference on Acoustic Speech and Signal Processing (ICASSP 2012), Kyoto. 2012. Dostupné z https://www.cs.toronto.edu/~hinton/absps/icassp12_dbn.pdf

[13] Scientists See Promise in Deep-Learning Programs. *The New York Times* [online]. 2012 [cit. 2014-05-10]. Dostupné z: http://www.nytimes.com/2012/11/24/science/scientists-see-advances-in-deep-learning-a-part-of-artificial-intelligence.html

[14] DAHL, G. E., T. N. SAINATH a G. E. HINTON. *Improving Deep Neural Networks for LVCSR Using Rectified Linear Units and Dropout* In IEEE International Conference on Acoustic Speech and Signal Processing (ICASSP 2013) Vancouver, 2013 Dostupné z https://www.cs.toronto.edu/~hinton/absps/georgerectified.pdf

[15] BERGSTRA, J., O. BREULEUX, F. BASTIEN, P. LAMBLIN, R. PASCANU, G. DESJARDINS, J. TURIAN, D. WARDE-FARLEY a Y. BENGIO. Theano: A CPU and GPU Math Expression Compiler. *Proceedings of the Python for Scientific Computing Conference (SciPy) 2010*. June 30 - July 3, Austin, TX. Dostupné z http://www.iro.umontreal.ca/~lisa/pointeurs/theano_scipy2010.pdf

[16] DeepLearning 0.1 documentation: Deep Learning Tutorials. LISA LAB. *Deep Learning* [online]. 2008 [cit. 2013-10-23]. Dostupné z: http://www.deeplearning.net/tutorial/

[17] SEIDE, F., G. LI, a D. YU. *Conversational speech transcription using context-dependent deep neural networks*, in Proc. Interspeech, 2011, s. 437–440. Dostupné z: http://research.microsoft.com/pubs/153169/CD-DNN-HMM-SWB-Interspeech2011-Pub.pdf

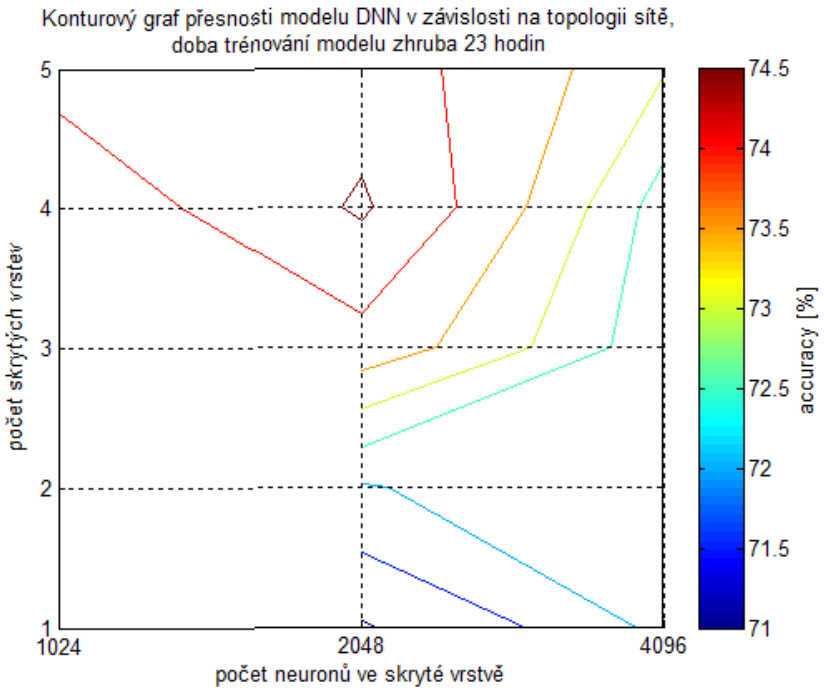
[18] JAITLY, N., P. NGUYEN, A. SENIOR, a V. VANHOUCKE, *An application of pretrained deep neural networks to large vocabulary speech recognition*, in Proc.Interspeech, 2012. Dostupné z: http://www.cs.toronto.edu/~ndjaitly/techrep.pdf

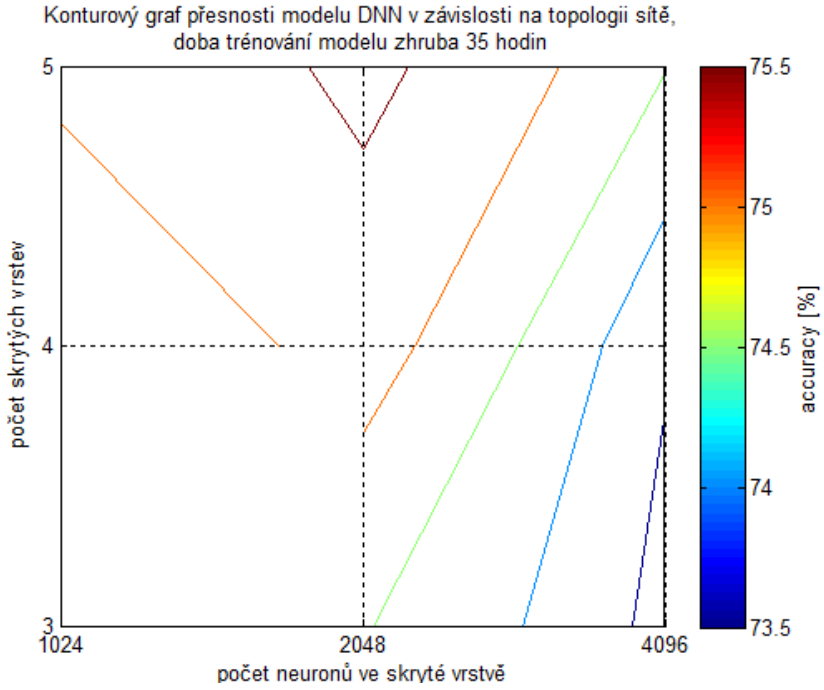
Přílohy

A Obsah přiloženého CD

- adresář *instalace* obsahuje některé soubory nutné pro instalaci softwarového toolkitu
- adresář *nejlepší model* obsahuje akustický model neuronové sítě s nejvyšší přesností
- soubor *vysledkyExperimentu.zip* obsahuje veškeré trénovací logy a soubory s výsledky rozpoznávání řeči
- soubor *diplomovaPrace.pdf* obsahuje elektronickou podobu této práce

B Konturové grafy přesnosti podle topologie sítě





C Pro testovací sadu Broadcast

* označeny sítě modelované na druhé pracovní stanici

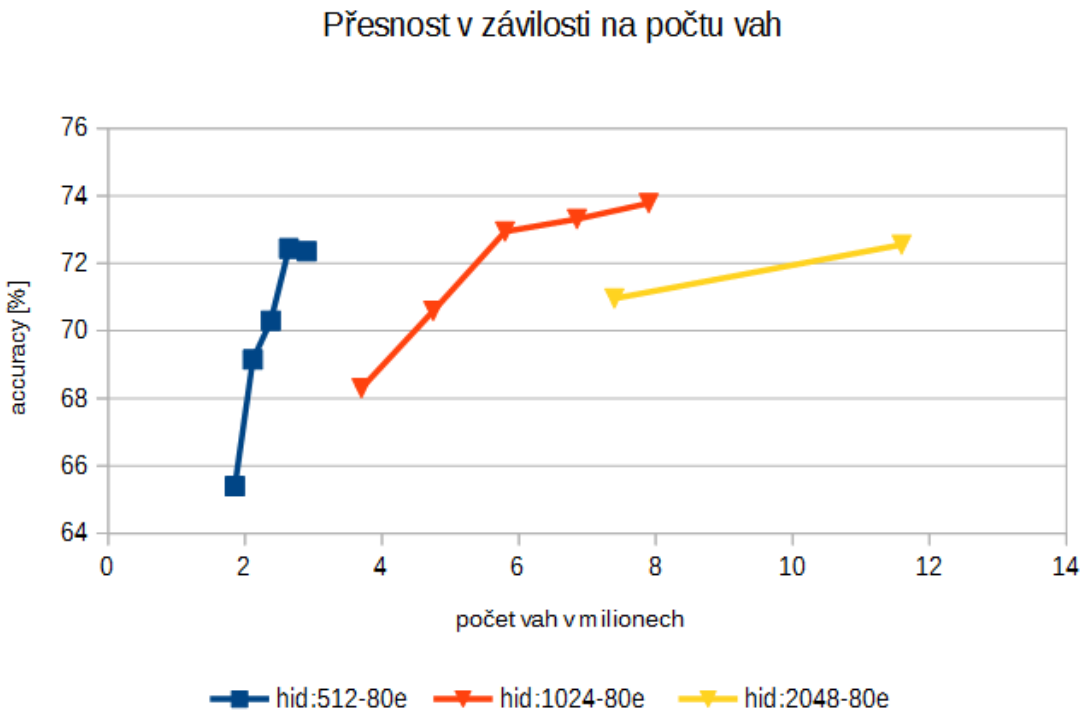
Přesnost podle topologie sítě (770 minut) [%]				
počet vrstev	počet neuronů ve skryté vrstvě			
	512	1024	2048	4096
1	66,26*	67,91	69,17	69,95
2	69,15	70,05	70,26	69,5
3	69,72*	71,78	70,94*	69,79
4	71,78	71,89*	73,14	69,29
5	71,79*	72,38*	72,51*	70,05
6			72,92	
7			71,93*	
8			72,09	
9			69,23*	
10			70,59	

Přesnost podle topologie sítě (1400 minut) [%]				
počet vrstev	počet neuronů ve skryté vrstvě			
	512	1024	2048	4096
1			70,96	72,14
2			71,97	72,42
3		73,89	73,82*	72,11
4		73,72*	74,57	72,26
5	73,31*	74,13*	74,24*	73,06
6			74,4	
7			73,7*	
8			74,32	
9			71,79*	
10			73,56	

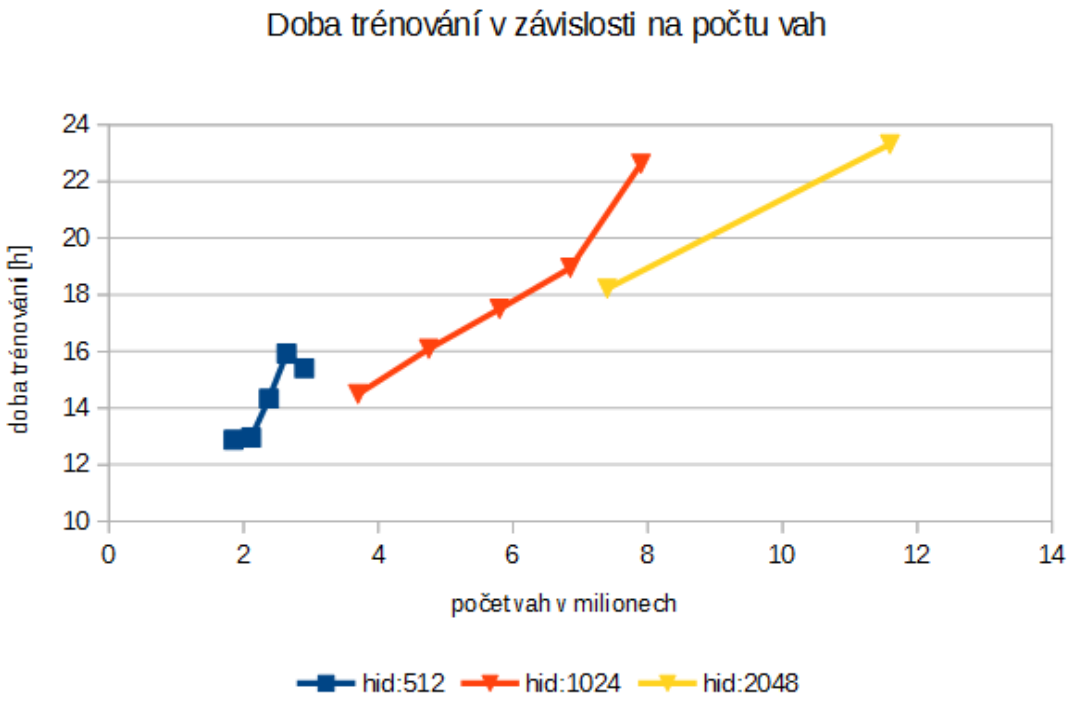
Přesnost podle topologie sítě (2100 minut) [%]				
počet vrstev	počet neuronů ve skryté vrstvě			
	512	1024	2048	4096
1				
2				73,08
3			74,53*	73,33
4		74,61*	75,21	73,56
5		75,1*	75,62*	74,53
6			75,41	
7			74,65*	
8			75,8	
9			73,33*	
10			74,4	

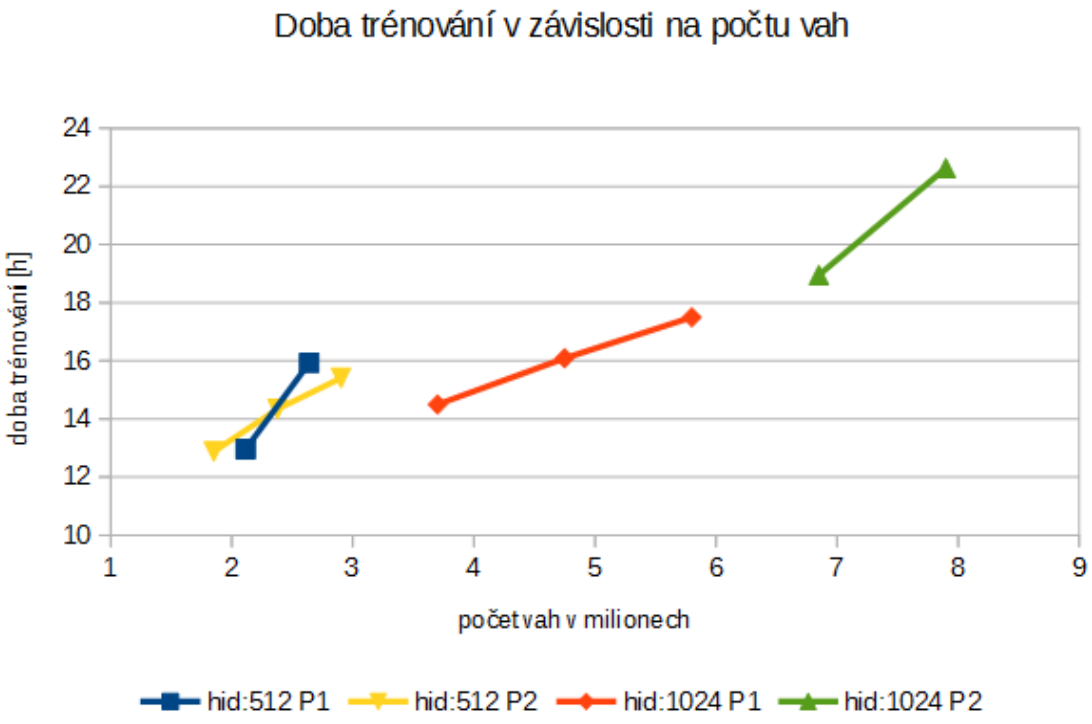
D Přesnost v závislosti na počtu vah pro sítě široké 512, 1024 a 2048 neuronů

Trénování sítí bylo ukončeno po uplynutí 80 epoch. Počet vrstev byl znázorněn počtem bodů na úsečce zleva.

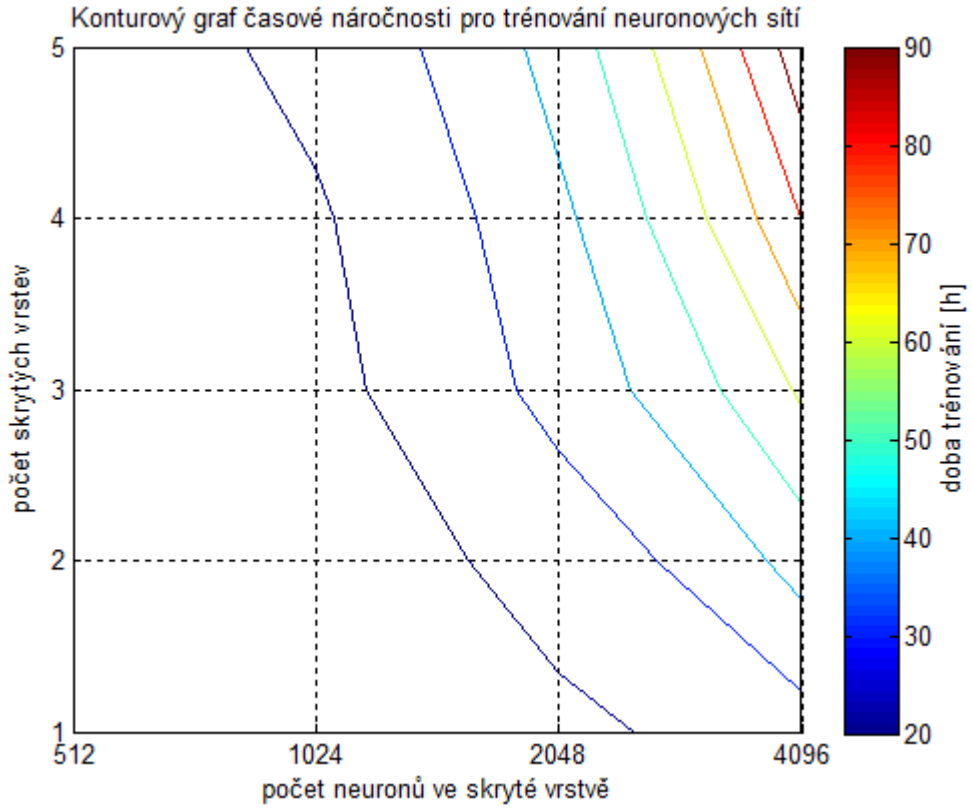


E graf pro 80 epoch P1 prac stanice 1 P2 prac stanice 2

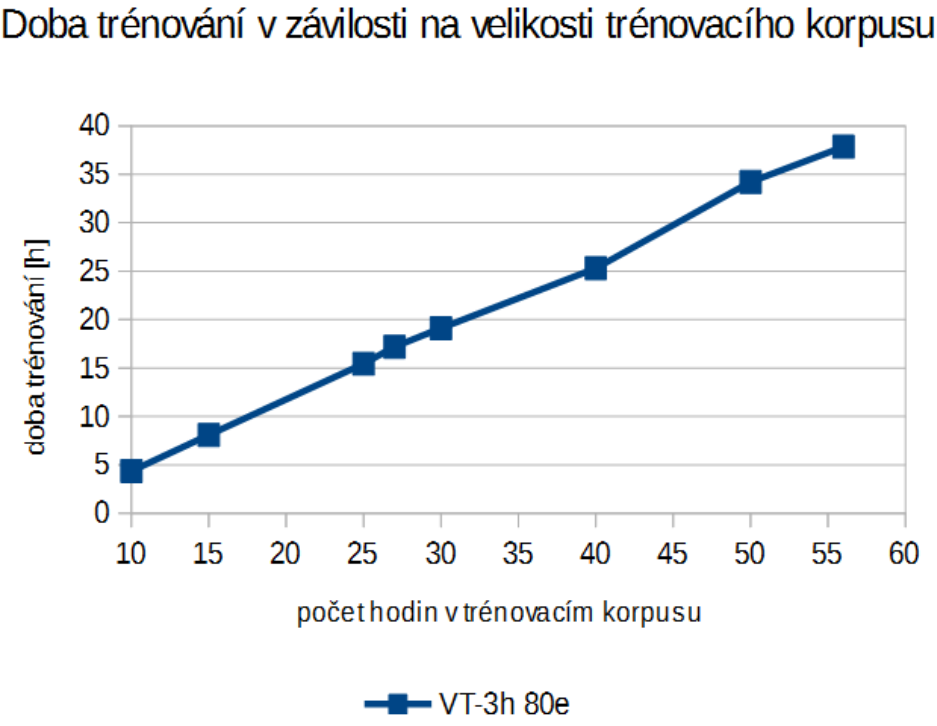


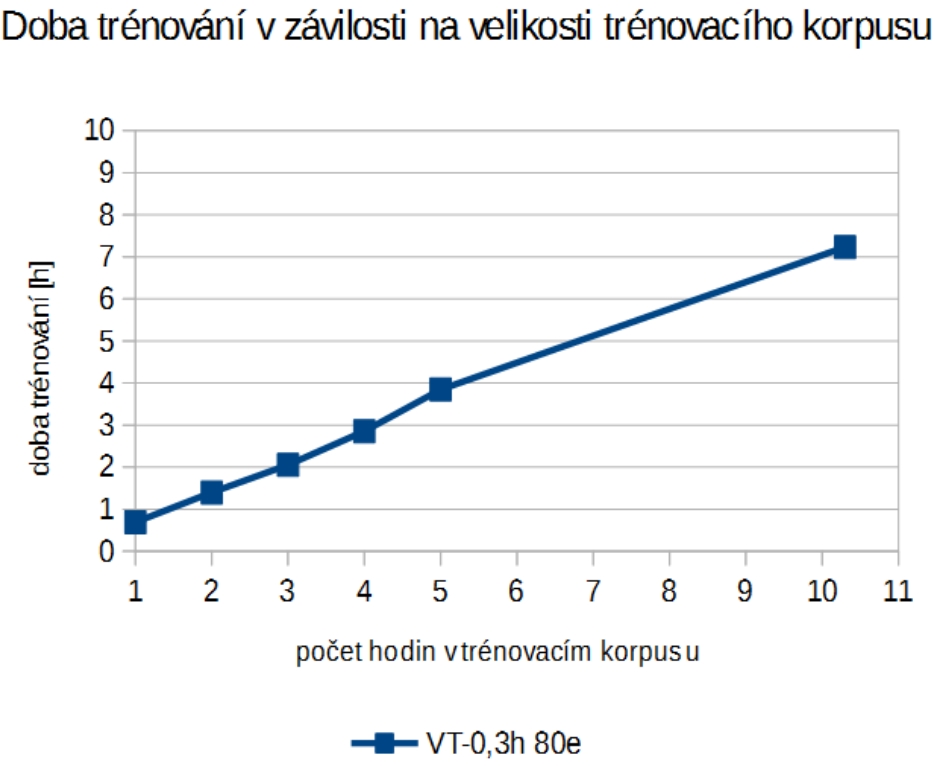


F Konturový graf časové náročnosti trénování pro 80 epoch



G doba trénování vzhledem k množství dat v trénovacím korpusu





H Výsledky k experimentů s předtrénováním neuronové sítě

Diskriminativní předtrénování po vrstvách 1 počet vrstev: 5 šířka sítě: 1024								
Počet epoch	40	60	80	100	111	120	160	200
Doba před + trén [h]	69,49	74,23	78,97	83,71	86,32	88,46	97,94	107,43
Broadcast acc [%]	75	75,49	75,76	75,74	75,66	75,47	**75,85**	75,95
Justice acc [%]	90,57	90,66	90,65	90,52	90,55	90,53	**90,64**	90,55
Pomocne acc [%]	91,76	91,21	91,48	91,21	90,66	90,66	**91,48**	90,93
Arit. průměr acc [%]	85,78	85,79	85,96	85,82	85,62	85,55	**85,99**	85,81

Diskriminativní předtrénování po vrstvách 2 počet vrstev: 5 šířka sítě: 1024							
Počet epoch	40	80	120	160	186	200	240
Doba před + trén [h]	14,59	29,18	43,77	58,36	67,85	72,95	87,54
Broadcast acc [%]	72,88	74,9	76,18	76,48	**76,75**	76,3	76,24
Justice acc [%]	89,9	90,4	90,49	90,43	**90,4**	90,47	90,6
Pomocne acc [%]	90,11	92,58	92,03	92,03	**92,31**	92,31	92,03
Arit. průměr acc [%]	84,30	85,96	86,23	86,31	**86,49**	86,36	86,29

Žádné předtrénování počet vrstev: 5 šířka sítě: 1024															
Počet epoch	20	40	60	80	140	166	180	220	260	300	340	380	420	460	500
Doba trénování [h]	5,32	10,64	15,97	21,29	37,25	44,17	47,90	58,54	69,19	79,83	90,47	101,12	111,76	122,41	133,05
Broadcast acc [%]	68,76	72,11	73,17	73,78	75,51	76,53	76,46	76,57	76,73	**76,81**	77,18	77,23	77	76,94	76,79
Justice acc [%]	89,23	89,86	90,11	90,25	90,49	90,69	90,74	90,83	90,84	**90,89**	90,72	90,7	90,47	90,54	90,41
Pomocne acc [%]	84,34	87,91	89,56	89,56	91,21	90,66	91,48	91,48	90,11	**92,03**	90,93	90,93	90,66	92,03	92,03
Arit. průměr acc [%]	80,78	83,29	84,28	84,53	85,74	85,96	86,23	86,29	85,89	**86,58**	86,28	86,29	86,04	86,50	86,41

I Výsledky k experimentů s topologií sítě

Testovací korpus PL Broadcast

Baseline 73,14%

Acc. pro 512 neuronů v s. v. [%]				
vrstvy	epochy			
	20	40	60	80
1	59,54	62,65	64,3	65,4
2	61,87	66,71	68,34	69,15
3	63,05	67,66	69,25	70,3
4	66,63	69,44	71,33	72,44
5	66,53	69,6	71,33	72,36

Acc. pro 1024 neuronů v s. v. [%]				
vrstvy	epochy			
	20	40	60	80
1	62,8	65,77	67,48	68,3
2	64,55	68,38	69,95	70,59
3	66,78	69,93	71,78	72,94
4	68,08	71,1	72,44	73,31
5	68,76	72,11	73,17	73,78

Acc. Pro 2048 neuronů v s. v. [%]				
vrstvy	epochy			
	20	40	60	80
1	65,48	67,91	69,62	70,96
2	66,76	69,93	71,66	72,55
3	68,82	71,64	73,37	74,11
4	70,82	73,52	74,55	75,12
5	71,43	73,74	74,96	75,52
6	71,76	74,28	75,35	75,97
7	71,11	73,66	74,55	75,33
8	72,57	74,75	75,82	76,46
9	70,47	73,04	74,26	75,14
10	71,74	74,4	75,51	76,17

Acc. Pro 4096 neuronů v s. v. [%]				
vrstvy	epochy			
	20	40	60	80
1	66,63	69,95	71,41	72,36
2	68,73	71,93	73,04	73,56
3	70,16	73,02	74,01	74,61
4	71,83	73,78	74,55	75,16
5	73,06	74.77	75,66	76,52

Acc. pro 81 epochu celk. přehled				
vrstvy	počet neuronů			
	512	**1024**	**2048**	**4096**
1	65,4	68,3	70,96	72,36
2	69,15	70,59	72,55	73,56
3	70,3	72,94	74,11	74,61
4	72,44	73,31	75,12	75,16
5	72,36	73,78	75,52	76,52
6			75,97	
7			75,33	
8			76,46	
9			75,14	
10			76,17	

Testovací korpus PL Justice

Baseline 88,09%

Acc. pro 512 neuronů v s. v. [%]				
vrstvy	epochy			
	20	40	60	80
1	85,75	87,1	87,51	87,73
2	86,06	87,57	88,49	88,7
3	87,33	88,68	89,33	89,58
4	87,51	88,84	89,1	89,49
5	88,18	89,23	89,71	90,02

Acc. pro 1024 neuronů v s. v. [%]				
vrstvy	epochy			
	20	40	60	80
1	86,52	87,4	87,96	88,37
2	87,15	88,31	88,84	88,96
3	87,88	89,19	89,62	89,81
4	88,68	89,61	90	90,27
5	89,23	89,86	90,11	90,25

Acc. Pro 2048 neuronů v s. v. [%]				
vrstvy	epochy			
	20	40	60	80
1	87,1	87,86	88,3	88,63
2	87,87	88,88	89,26	89,48
3	88,44	89,42	89,67	89,85
4	89,02	89,69	90,07	90,25
5	89,23	89,75	90,53	90,7
6	89,49	90,03	90,32	90,37
7	89,71	90,21	90,49	90,54
8	89,25	89,87	90,1	90,25
9	89,47	90,21	90,56	90,44
10	88,97	89,75	90,03	90,23

Acc. Pro 4096 neuronů v s. v. [%]				
vrstvy	epochy			
	20	40	60	80
1	87,28	88,16	88,51	88,6
2	88,58	89,24	89,53	89,52
3	88,87	89,71	89,92	90,04
4	89,34	89,73	89,98	90
5	89,61	90,15	90,05	90,11

Acc. pro 81 epochu celk. přehled				
vrstvy	počet neuronů			
	512	**1024**	**2048**	**4096**
1	87,73	88,37	88,63	88,6
2	88,7	88,96	89,48	89,52
3	89,58	89,81	89,85	90,04
4	89,49	90,27	90,25	90
5	90,02	90,25	90,7	90,11
6			90,37	
7			90,54	
8			90,25	
9			90,44	
10			90,23	

Testovací korpus PL Pomocne

Baseline 85,16%

Acc. pro 512 neuronů v s. v. [%]				
vrstvy	epochy			
	20	40	60	80
1	78,3	80,22	82,14	82,42
2	81,32	83,79	85,99	86,81
3	80,22	85,71	86,54	87,09
4	85,16	85,16	87,64	88,74
5	84,89	85,99	88,74	88,74

Acc. pro 1024 neuronů v s. v. [%]				
vrstvy	epochy			
	20	40	60	80
1	79,95	83,52	84,34	85,44
2	82,14	85,71	87,91	87,64
3	84,07	87,09	88,46	89,56
4	84,62	88,46	89,56	89,56
5	84,34	87,91	89,56	89,56

Acc. Pro 2048 neuronů v s. v. [%]				
vrstvy	epochy			
	20	40	60	80
1	81,04	84,34	84,89	85,16
2	84,62	87,91	89,56	90,66
3	86,54	89,01	89,84	89,84
4	88,74	91,21	90,93	91,48
5	89,29	90,93	91,21	92,03
6	87,91	89,01	89,84	89,56
7	89,29	89,56	91,21	91,76
8	88,19	90,93	90,93	90,66
9	87,64	90,38	91,21	92,03
10	86,81	89,84	91,21	91,48

Acc. Pro 4096 neuronů v s. v. [%]				
vrstvy	epochy			
	20	40	60	80
1	82,42	86,26	86,54	87,64
2	85,71	87,64	88,46	89,56
3	87,09	89,29	89,84	90,11
4	88,46	90,66	90,93	91,76
5	89,56	91,48	91,76	92,03

Acc. pro 81 epochu celk. přehled				
vrstvy	počet neuronů			
	512	**1024**	**2048**	**4096**
1	82,42	85,44	85,16	87,64
2	86,81	87,64	90,66	89,56
3	87,09	89,56	89,84	90,11
4	88,74	89,56	91,48	91,76
5	88,74	89,56	92,03	92,03
6			89,56	
7			91,76	
8			90,66	
9			92,03	
10			91,48	

J Postup přípravy pracovní stanice pro trénování DNN (Microsoft Windows 7)

1) Nainstalovat Microsoft Windows SDK.
http://www.microsoft.com/en-us/download/details.aspx?id=8279

2) Rozbalit MinGW z přílohy na CD.
Přidat složku MinGW\bin do *path* v *Proměnném prostředí*.

3) Nainstalovat CUDA 5.0.
https://developer.nvidia.com/cuda-toolkit
Nastavit a upravit cesty v *nvcc.profile* umístěném v (příklad na CD):
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v5.0\bin.

4) Nainstalovat python 2.7.6 pro 64bit.
https://www.python.org/
Přepsat složku *Python27* rozbalenou přílohou *Python27.7z* z CD.
Upravit cesty v .theanorc a zkopírovat do uživatelské složky např. C:\Users\Martin\.

5) Nainstalovat IDE, např. VS 2010 nebo Apthana studio.

6) Při výpočtech na GPU nepoužívat program *Připojení ke vzdálené ploše*
(také využívá GPU a koliduje).

(příklad jak by mělo vypadat *path* v *Proměnném prostředí* na CD)

V případě jiného operačního systému viz http://deeplearning.net/software/theano/install.html